Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра Програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліни “Бази даних”

“Інформаційна система «Яхт-клуб»”

Керівник: доц. каф. ПІ Мазурова О.О.

Студент гр. ПІ - 15 – 1 Пироженко С. С.

Комісія:

Доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Мазурова О.О.

Ст. викл. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Черепанова Ю.Ю. Ст. викл. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Широкопетлєва М.С.

Харків 2016

Харківський національний університет радіоелектроніки

**Кафедра** \_\_\_Програмної інженерії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Дисципліна** \_\_\_Бази даних \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Спеціальність** \_Програмна інженерія\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Курс** \_\_\_2\_\_\_\_\_\_\_ **Група** \_\_\_ПІ-15-1\_\_\_\_\_\_\_ **Семестр**\_\_\_\_ 3\_\_\_\_\_\_\_\_

**ЗАВДАННЯ**

**на курсову роботу студента**

*Пироженка Сергія Станіславовича*

**1. Тема роботи:** «Інформаційна система «*Яхт-клуб* »»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

**2. Строк здачі закінченої роботи** \_\_\_23.12.2016\_\_\_\_\_\_\_\_

**3. Вихідні дані для роботи:** методичні вказівки до виконання курсової роботи, вимоги до інформаційної системи, предметна область, що пов’язана *з обліком архітектурних споруд.*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**4. Зміст розрахунково - пояснювальної записки:** вступ, аналіз предметної області; постановка задачі; проектування бази даних; опис програми; висновки; перелік посилань.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5. Перелік графічного матеріалу:** загальна схема концептуальної моделі, ER-діаграма, структура 1НФ, 2НФ, 3НФ, схема БД в 3НФ, UML-діаграми, копії екранів (“скріншоти”) прикладної програми, приклади звітів прикладної програми\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6. Дата видачі завдання** \_\_\_09.09.16 р.\_\_\_

КАЛЕНДАРНИЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Номер | Назва етапів курсової роботи | Строк виконання етапів роботи | Примітки |
| 1 | Аналіз предметної області | 9.09.16 – 24.09.16 | Виконано |
| 2 | Постановка задачі | 20.09.16 – 30.09.16 | Виконано |
| 3 | Побудова ER-діаграми бази даних | 27.09.16 – 15.10.16 | Виконано |
| 4 | Оформлення розділів 1, 2 та 3.1, 3.2 пояснювальної записки | 15.10.16 - 27.10.16 | Виконано |
| 5 | Перша контрольна точка з курсового проекту | 24.10.16 – 28.10.16 | Виконано |
| 6 | Нормалізація бази даних | 20.10.16 - 10.11.16 | Виконано |
| 7 | Створення демо-версії програми | 20.10.16 – 20.11.16 | Виконано |
| 8 | Тестування програми, наповнення бази даних | 15.11.16 - 25.11.16 | Виконано |
| 9 | Друга контрольна точка з курсового проекту | 21.12.16– 02.12.16 | Виконано |
| 10 | Реалізація остаточної версії програми | 1.12.16-15.12.16 | Виконано |
| 11 | Оформлення інших розділів пояснювальної записки | 1.11.16 – 15.12.16 | Виконано |
| 12 | Захист курсового проекту (третя контрольна точка) | 12.12.16- 27.12.16 | Виконано |

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Керівник \_\_\_\_ *доц. Мазурова О.О..*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2016 р.

РЕФЕРАТ

Пояснювальна записка до курсової роботи: с., рис., додаток, джерел.

Метою роботи є розробка програми “Яхт-клуб” використовуючи ООП та проектуючи базу даних для програми.

Методи розробки базуються на використанні середи розробки Microsoft Visual Studio 2015, мови програмування C#, мови запитів SQL та реляційної бази даних.

У результаті отримана інформаційна система “Яхт-клуб”, яка дозволяє користувачу маніпулювати інформацією, виконувати пошук, фільтрацію, сортування, додавання, редагування та видалення даних. У системі “Яхт-клуб” також є можливість отримувати звіти та статистику.

Посилання на github:

ПРОГРАМА, ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, БАЗА ДАНИХ, КЛАС, SQL, МОВА ПРОГРАМУВАННЯ С#, ER-ДІАГРАМА, СТАТИСТИКА, КОНЦЕПТУАЛЬНА МОДЕЛЬ, ЯХТА, ТАБЛИЦЯ, НОРМАЛІЗАЦІЯ, ЯХТ-КЛУБ, ІНТЕРФЕЙС, АВТОМАТИЗАЦІЯ.

ЗМІСТ

1. Аналіз та концептуальне моделювання предметної області…………………………7

1.1 Аналіз предметної області…………………………………………………………….7

1.2 Концептуальне моделювання предметної області……………..................................8

2. Постановка задачі………………………………………………………………………15

3. Проектування бази даних……………………………………………………...……....17

3.1 UML-моделювання………………………………………………………...………… 17

3.2 Побудова ER-діаграми……………………………………………..............................18

3.3 Побудова схеми реляційної бази даних у третій нормальній формі………….......19

4. Опис програми……………………………………………………………………….…26

4.1 Загальні відомості …………………………………………………………………….26

4.2 Виклик і завантаження…………………………………………………………….….26

4.3 Призначення і логічна структура…………………………………………………….26

4.4 Опис фізичної моделі бази даних………………………………………………….…27

4.5 Опис програмної реалізації…………………………………………………………...30

4.6 Опис задачі автоматизації ……………………………………………………………35

5. Висновки………………………………………………………………………………...42

6. Перелік посилань………………………………………………………………………..43

ВСТУП

Сьогодні яхта це не тільки засіб пересування, а в першу чергу, елемент, що характеризує достаток і успішність. Найдешевші екземпляри можуть дозволити собі далеко не всі, однак, мабуть всі люблять активний відпочинок на воді і хоч раз в житті мріяли здійснити подорож на яхті, адже це можливість побувати в різних місцях, побачити новий світ. Окрім того, яхти для сучасного світу стали одним із способів вкладення матеріальних коштів, а також самовираження.

В наш час збільшення кількості яхт привело до бурного розвитку яхт клубів. Кожен з них має свій, відмінний від інших устав та установчі документи і звичайна людина часто не може вірно зорієнтуватись та вибрати те, що йому потрібно.

Вищезазначена потреба має глобальний характер, тому потрібно задовольнити її якнайшвидше. Для цього необхідно створити єдину комп’ютерну систему, яка б містила в собі основні дані про яхт-клуби, яхти та дозволяла користувачу придбати яхту своєї мрії, а також відправитись на ній в подорож.

Метою даної курсової роботи є створення інформаційної системи «Яхт-клуб», яка б дозволила вирішувати описані перед цим проблеми. Дана система буде реалізована як десктопна програма, що буде працювати на абсолютно всіх комп’ютерах. Програмний продукт буде реалізований за допомогою SQL Server Express LocalDB, що вбудований у середу розробки Visual Studio за допомогою мови програмування C#.

В результаті виконання курсової роботи, ми отримуємо інформаційну систему, що містить в собі інформацію яхти, яхт-клуби, їх власників та порти. Система буде давати можливість отримувати, додавати, редагувати, видаляти інформацію, сортувати та фільтрувати її, проводити пошук. Також система надасть можливість користувачам, що придбали яхти відправитися у подорож за допомогою менеджера подорожей.

1. АНАЛІЗ ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області

У наш час яхти можуть собі дозволити лише найбільш успішні люди. Деякі з цих людей захоплюються вітрильним спортом або просто хочуть знайти людей зі спільними інтересами, саме тому ще у 17 столітті почали з’являтися яхт-клуби. Яхт-клуб — громадська чи приватна організація, що об'єднує яхтсменів з стягненням або без щорічних внесків до фонду клубу. Яхт-клуби націлені на збереження та охорону яхт, прав “яхтсменів”. Комплекс споруд яхт-клубів включає елінг, майстерні, причали для швартування та стоянки яхт, механізми для підйому і спуску суден. Вони розташовуються зазвичай в бухті або гирлі річки. Окремі яхт-клуби проводять змагання. У кожного яхт-клубу, зазвичай, є власник, що й організовує змагання, наймає людей, що надають можливість власникам яхт, що вступили до яхт-клубу, користуватися різними сервісами, такими, як купівля, продаж, обмін, ремонт та інші.

Через те, що власники яхт-клубів повинні вести певну статистику та звіти про, наприклад, щорічні розходи та мати можливість організовувати змагання, про які зможе дізнатися кожен, було прийнято рішення про створення комп’ютерної інформаційної системи «Яхт-клуб». Ця система буде мати у собі інформацію про яхт-клуби, що, у свою чергу, будуть пропонувати різні сервіси та змагання. ІС також надаватиме можливість редагувати, додавати інформацію про яхт-клуби, яхти та їх власників.

Інформаційна система «Яхт-клуб» буде надавати можливість маніпулювати інформацією про яхти, їх власників, а саме: додавати нових власників та їх яхти, видаляти, редагувати. Також власники яхт матимуть можливість купувати та продавати яхти, а також система буду надавати можливість організувати безпечну подорож.

Інформаційна система «Яхт-клуб» необхідна для надання інформації про різні яхт-клуби, також вона надаватиме можливість власникам яхт продавати свою власність та купувати нові яхти.

Під час роботи у системі буду використовуватися наступна інформація:

* інформація про яхт-клуб: назва клубу, власник (якщо він є), дата створення, місцезнаходження (порт чи бухта), гроші на рахунку;
* інформація про яхти: назва, власник та його дані (якщо вона вже привласнена), рік випуску, різні характеристики, а саме: довжина, висота, ширина, потужність та інше), інформація про те, чи знаходиться ця яхта в порту;
* інформація про власників яхт: ПІБ, вік, місце проживання, сума на рахунку;
* інформація про продаж, купівлю певної яхти.

1.2 Концептуальне моделювання предметної області

1.2.1 Опис функціональної структури системи

Для опису функціональної структури ІС використовується USE-CASE діаграма (діаграма прецедентів) наведена нижче, яка відображає відношення між акторами та прецедентами. (див. рис. 1.1).

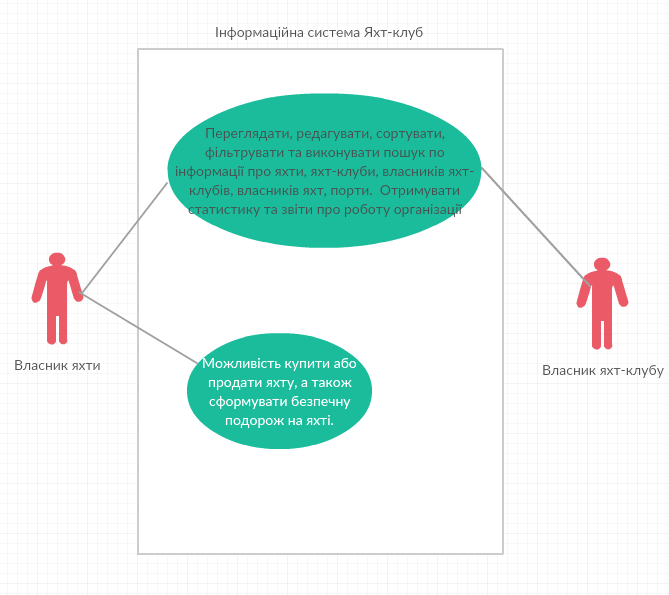


Рисунок 1.1 – USE-CASE діаграма ІС «Яхт-клуб»

1.2.2 Опис об'єктів та зв'язків між ними

В інформаційну систему «Яхт-клуб» закладені такі 5 основних об’єктів: яхта, яхт-клуб, власник яхт-клубу, власник яхти, порт.

Усі вищезазначені сутності пов’язані між собою наступним чином (див. рис. 1.2):

* власник має одну або декілька яхт;
* власник яхти знаходиться у яхт-клубі;
* яхт-клуб знаходиться у порту або бухті;
* яхт-клуб має власника;
* яхта може мати власника або ні;
* яхта може перебувати в порті або ні.

База даних ІС «Облік творів скульптури» повинна містити у собі наступну інформацію:

* інформація про яхту: назва, тип, рік випуску, висота, висота до палуби, ширина, довжина, ціна, розміщення (у порту або ні), номер власника ( привласнена або у продажу);
* інформація про власника яхти: ім’я, вік, адреса, рахунок, яхт-клуб у якому він перебуває;
* інформація про яхт-клуб: назва, рахунок, власник, порт, у якому знаходиться клуб;
* інформація про власника яхт-клубу: ім’я, вік, адреса, рахунок;
* інформація про порт: назва, місце дислокації.

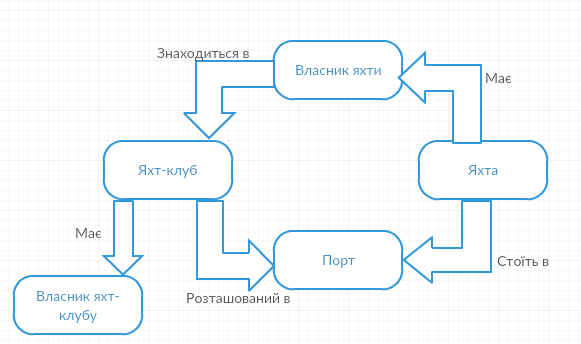


Рисунок 1.2 – Об’єкти та зв’язки між ними

1.2.3 Опис інформаційних потреб користувачів

ІС «Яхт-клуб» має наступних потенційних користувачів:

1. власник яхт-клубу потребує повну інформацію щодо власників яхт, що знаходяться у його клубі, порту або бухти, де розташований його яхт-клуб. Повинен мати можливість редагувати свою інформацію, інформацію свого яхт-клубу.
2. Переглядати інформацію з можливістю:
   1. пошуку яхт за всіма атрибутами або за обраним (власник, назва, ціна, місцезнаходження);
   2. сортування яхт за назвою, ціною тощо.
   3. Фільтрація за ціною, типом тощо.
3. Отримувати статистику та звіти
   1. статистика про яхти по дислокації (кількість, загальна ціна яхт);
   2. статистика про найпопулярніший тип яхт (вид, кількість, середня ціна по виду);
   3. статистика про власників яхт (середній вік, середня кількість грошей на рахунку, загальна кількість грошей на рахунку, найпопулярніший яхт-клуб);
   4. статистика про яхти по виду власності (кількість, ціна);
   5. звіт про кількість власників, що вступили до кожного яхт-клубу ( назва клубу, кількість);
   6. звіт деякої інформації по обраному клубу ( яхт клуб, назва, місце розташування, рахунок клубу, кількість учасників, їх середній вік та сума грошей на рахунках).
4. Власник яхти має можливість отримувати інформацію про яхти та яхт. Власник може прокладати маршрут для безпечної подорожі на обраній яхті. Він може редагувати дані про себе та яхту, продавати яхту, а також купувати нову.

Робота автоматизації полягає у прогнозуванні безпечних місць для певної яхти, ґрунтуючись на її характеристиках (розмір, висота та інше) та типах (вітрильна, моторна та інше). Система буде за формулою визначати, чи підходить погода у даний момент (вітер, висота хвиль, опади) для яхти та покаже найкращі та безпечні місця для вашої подорожі. Нижче наведено більш детально.

При вході власник яхти, у разі небезпеки, отримує інформацію від метеостанцій про небезпечні місця, що знаходяться поблизу нього. Власник яхти може подивитися про це більш детально, наприклад, дізнатися про приблизну тривалість цієї небезпеки.

Коротко про програмну реалізацію: метеостанція по даним про швидкість вітру визначає висоту хвиль за формулою : (x \* 0.2) \* i = висота хвилі, де

x – швидкість вітру в метрах на секунду;

i – коефіцієнт таблиці Бофорта ( більш детально у пункті 2.2.6)

У подальшому, система отримує доступ до інформації про яхти, що знаходяться неподалеку, дивиться характеристики яхт, і , у разі якщо їм загрожує небезпека ( наприклад, для вітрильних яхт – висока швидкість вітру або , доречно для будь-якої яхти, висота хвиль занадто висока), відправляє повідомлення власникам яхт.

Залежність висоти яхти від висоти хвилі, що загрожує їй, наведена у пункті 1.2.6.

Якщо висота хвилі більша ніж ¾ висоти від   
“рівня” до палуби, то така хвиля має змогу перевернути яхту, якщо її висота вище палуби більше ніж висота від “рівня” до палуби помноженої на 2.

1.2.4 Опис документообігу

Документообіг ІС «Яхт-клуб» містить в собі статистику та звіти щодо яхт-клубу, погодних умов тощо.

Система має змогу відображати таку статистику:

* статистика про яхти по місцю дислокації ( кількість, загальна ціна яхт);
* статистика про найпопулярніший тип яхт (вид, кількість, середня ціна по виду);
* статистика про власників яхт ( середній вік, середня кількість грошей на рахунку, загальна кількість грошей на рахунку, найпопулярніший клуб);
* статистика про яхти по виду власності ( кількість, ціна).

Система має можливість роздрукувати наступні звіти:

* звіт про кількість власників, що вступили до кожного яхт-клубу (назва клубу, кількість);
* звіт деякої інформації по обраному клубу ( яхт клуб, назва, місце розташування, рахунок клубу, кількість учасників, їх середній вік та сума грошей на рахунках).

1.2.5 Обмеження цілісності

Обмеження щодо первинних ключів об’єктів:

* різні яхти можуть мати однакові назви, тому для ідентифікації яхти в базі даних їй надається унікальний номер;
* різні власники можуть мати однакові імена, тому для спрощення ідентифікації власника в базі даних кожному з них надається унікальний номер;
* різні яхт-клуби можуть мати однакові назви, тому для ідентифікації в базі даних кожному клубу надається унікальний номер.
* різні порти можуть мати однакові назви, тому для ідентифікації в базі даних кожному порту надається унікальний номер.
* Різні власники яхт-клубів можуть мати однакові імена, тому для спрощення ідентифікації власника клубу в базі даних кожному з них надається унікальний номер;

Обмеження щодо зв’язків між об’єктами:

* кожна яхта має лише одного власника, але може бути орендована на будь-який проміжок часу;
* у кожному яхт-клубі може перебувати безліч власників яхт;
* яхт-клуб може мати лише одного власника;
* яхт-клуб може одночасно знаходитися лише в одному порту;
* в одному порту може знаходитися декілька яхт-клубів;
* яхт-клуб може продавати безліч яхт;
* кожна людина може мати безліч яхт;
* яхта одночасно може перебувати лише в одному порті або ж не перебувати зовсім;

1.2.6 Опис алгоритмічних залежностей

ІС «Яхт-клуб» має наступні алгоритмічні залежності:

* обчислення висоти хвилі від вітру:

За шкалою Бофорта можна визначити залежність висоти хвилі від вітру: (x \* 0.2) \* i = висота хвилі, де

x – швидкість вітру в метрах на секунду;

i – коефіцієнт таблиці Бофорта (i = 1.1, якщо швидкість вітру (х) = 0-5м/с; i = 1.2, якщо х = 5-10м/с i = 1.4, якщо х = 10-15м/с;

i = 1.7, якщо х = 15-20м/с; i = 2.2, якщо х > 20м/с);

* залежність висоти яхти від висоти хвилі, що загрожує їй:

Якщо висота хвилі більша ніж ¾ висоти від   
“рівня” до палуби, то така хвиля має змогу перевернути яхту, якщо її висота вище палуби більше ніж висота від “рівня” до палуби помноженої на 2. Для круїзних яхт, що мають висоту більше ніж 30м, ця формула втрачає сенс.

Залежності між одиницями виміру:

* для вимірювання довжини використовується морська миля:

1 морська миля (чверть) = 1852 м;

* швидкість яхт вимірюється у вузлах:

1 міжнародний вузол = 1 морська миля/година = 1,852 км/год.

1.2.7 Вимоги до програмної системи

Дана програмна система використовує СУБД MSSQL та розроблена як **”**Десктопна”програма.

Для даної запуску даної програми на вашому комп’ютері не потрібно ніяке додаткове програмне забезпечення.

2.2.8 Лінгвістичні відносини

Бухта **-** невелика частина моря, затоки, озера, водосховища, відокремлена від відкритих вод з трьох боків частинами суші (виступами берегів, скелями та прилеглими островами) і захищена ними від хвиль та вітру.

Метеорологíчна стáнція (метеостанція) — станція для проведення спостережень за погодою. Складена з метеомайданчика, на якому розташована більшість приладів, що фіксують метеоелементи, і замкненого приміщення, в якому встановлюється барометр, барограф і ведеться обробка спостережень. На бурях у відкритому морі та в ненаселених районах встановлюють автоматичні метеостанції. Одержані на метеостанціях дані кодують і надсилають до метеорологічних центрів.

1. ПОСТАНОВКА ЗАДАЧІ

Для кожної яхти зберігається наступна інформація: id, назва, id власника, ПІБ власника, рік випуску, стан, довжина, ширина, висота, тип, перебування в оренді, місцезнаходження.

Для кожного власника яхти зберігається наступна інформація: id, ПІБ, дата народження, місце проживання, кількість яхт, гроші на рахунку. Для кожного яхт-клубу зберігається інформація: id клубу, назва, власник, місцезнаходження, сума на рахунку. У одному клубі може бути безліч власників яхт, тому між цими сутностями існує зв'язок один до багатьох.

Кожен власник-яхт клубу має наступну інформацію: ПІБ, вік, місце проживання, гроші на рахунку.

Кожен яхт-клуб перебуває у порту або бухті. Яхта може знаходитися як у порту (бухті) так і перебувати у відкритому просторі.

У програмному продукті необхідно реалізувати наступне:

1. Відображення даних:

* відображення списку інформації про яхти;
* відображення списку інформації про власників яхт;
* відображення списку інформації про яхт-клуби;
* відображення списку інформації про власників яхт-клубів;
* відображення списку інформації про порти.

1. Пошук, фільтрація та сортування:

* пошук яхт, яхт-клубів, власників яхт, власників яхт-клубів, портів;
* пошук найближчої до власника яхти метеостанції;
* фільтрація інформації про власників яхт та яхт-клубів (за віком, грошима на рахунку);
* фільтрація інформації про яхти ( за ціною, типом, розмірами, роком);
* фільтрація інформації про яхт-клуби ( за грошима на рахунку);
* фільтрація інформації про порти ( за місцеположенням, на приклад, порти в Україні);
* сортування інформації про яхти, яхт-клуби, порти, власників яхт-клубів та яхт (за будь-яким полем, наприклад, сортування яхт за зростанням ціни).

1. Можливість редагувати інформацію про яхти, їх власників, яхт-клуби та інше.

* додавання інформації про яхти, яхт-клуби, порти, власників яхт-клубів та яхт;
* видалення виділеної інформації або інформації, що співпадає з заданим ідентифікатором для видалення, про яхти, яхт-клуби, порти, власників яхт, власників яхт-клубів.

1. Формування статистики та звітів:

* звіт «Інформація про кількість власників, що перебувають у кожному яхт-клубі»;
* звіт «Інформація за одним обраним клубом».
* статистика про яхти по місцю дислокації ( кількість, загальна ціна яхт);
* статистика про найпопулярніший тип яхт (вид, кількість, середня ціна по виду);
* статистика про власників яхт ( середній вік, середня кількість грошей на рахунку, загальна кількість грошей на рахунку, найпопулярніший клуб);
* статистика про яхти по виду власності ( кількість, ціна).

1. Автоматизація:

Робота автоматизації полягає у прогнозуванні безпечних місць для певної яхти, ґрунтуючись на її характеристиках (розмір, висота та інше) та типах (вітрильна, моторна та інше). Система буде за формулою визначати, чи підходить погода у даний момент (вітер, висота хвиль, опади) для яхти та покаже найкращі та безпечні місця для вашої подорожі.

Коли власник яхти вводить свій унікальний номер, він побачить список доступних яхт для подорожі. Обравши яхту, користувач системи повинен ввести в поле назву місця (або координати) куди він хоче відправитися. Далі користувач зможе побачить інформацію про свою яхту та про погодні умови в обраному регіоні, з подальшим повідомленням про доцільність цієї подорожі.

3.ПРОЕКТУВАННЯ БАЗИ ДАНИХ

3.1 UML-моделювання

Проаналізувавши модель інформаційної системи, можна зробити висновок, що у системі існує 2 актори:

1. Власник яхти має такі можливості як: перегляд, редагування, видалення, додавання, фільтрування, сортування даних. Також може отримувати статистики та звіт щодо кількості людей у різних яхт-клубах та вступати до них. Має можливість продати свою яхту або купити нову, а також організовувати безпечні подорожі за допомогою електронного менеджера подорожей.
2. Власник яхт-клубу має можливість перегляду: перегляд, редагування, видалення, додавання, фільтрування, сортування даних. Також може отримувати статистики та звіт щодо кількості людей у різних яхт-клубах та вступати до них. Має можливість отримувати звіти щодо роботи свого яхт-клубу.

Виходячи з вищесказаного можна побудувати діаграму прецедентів. (див. рис. 3.1).

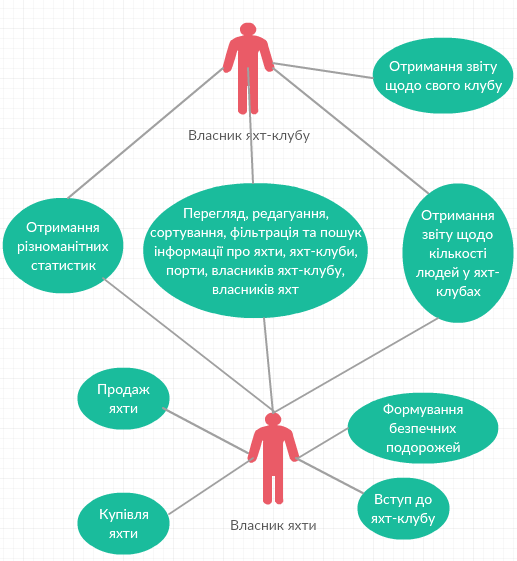


Рисунок 3.1 – діаграма прецедентів ІС «Яхт-клуб»

**3.2 Побудова ER-діаграми**

Проаналізувавши концептуальну модель предметної області, можна зробити висновок, що в базі даних ІС «Яхт-клуб» будуть наявні наступні об’єкти (сутності):

1. сутність «Яхта», що містить наступні атрибути: id, назва, тип, власник, рік випуску, висота, ширина, довжина, місцеположення, ціна.
2. сутність «Власник яхти», що містить наступні атрибути: id, ім’я, вік, адреса, рахунок, яхт-клуб ;
3. сутність «Яхт-клуб» має наступні атрибути: id, назва, власник, порт, рахунок;
4. сутність «Власник яхт-клубу» має наступні атрибути: id, ім’я, вік, адреса, рахунок ;
5. сутність «Порт» має наступні атрибути: id, місцеположення, назва;

Між вищенаведеними сутностями існують зв’язки:

1. між сутностями «Яхта» та «Власник яхти» зв’язок один-до-багатьох: один власник може мати багато яхт, одна яхта може мати лише одного власника;
2. між сутностями «Яхт-клуб» та «Власник яхт-клубу» зв’язок один-до-одного: один яхт-клуб може мати лише одного власників і ,навпаки, один власник може керувати лише одним яхт-клубом;
3. між сутностями «Яхт-клуб» та «Власник яхти» зв’язок один-до-багатьох: один яхт клуб може приймати до себе багато власників яхт, а один власник яхти може вступити лише до одного яхт-клубу скульптур;
4. між сутностями «Яхт-клуб» та «Порт» зв’язок багато-до-багатьох: один порт може вміщувати в собі багато яхт-клубів, але яхт-клуб може бути розміщений лише у одному порті.
5. між сутностями «Яхта» та «Порт» зв’язок один-до-багатьох: один порт може вміщувати в собі багато яхт, а одна яхта може стояти лише у одному порті;

На основі усього вищезазначеного можна побудувати ER-діаграма інформаційної системи «Яхт-клуб». (див. рис. 3.2)

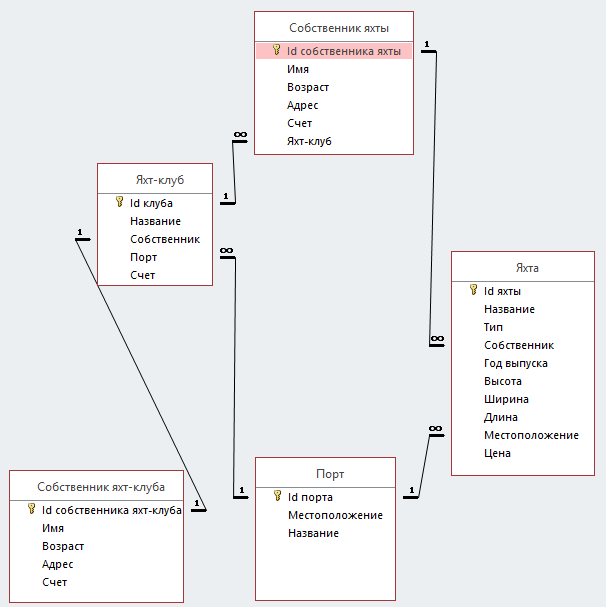


Рисунок 3.2 – ER**-**діаграма ІС «Яхт-клуб»

3.3 Побудова схеми реляційної бази даних у третій нормальній формі

Спочатку побудуємо схему бази даних у першій нормальній формі, визначимо первинні ключі та покажемо залежності між атрибутами (див. рис. 3.3).

Відношення знаходиться в 1НФ, якщо всі його атрибути є простими, всі використовувані домени повинні містити тільки скалярні значення. Не повинно бути повторень рядків таблиці.

Т1 ( 1 НФ)

\*Id яхти

Назва

Тип

Рік випуску

Висота

Висота до палуби

Ширина

Довжина

Ціна

Id Власника яхти

Ім’я

Вік

Адреса

Рахунок

Id яхт-клубу

Назва

Рахунок

Id порту

Місце розташування

Назва

Id власника клубу

Ім’я

Вік

Адреса

Рахунок

Рисунок 3.3 – Т1 у 1НФ

Для переходу до 2НФ необхідно, щоб кожен неключовий атрибут був у повній функціональній залежності від первинного ключа. Вище наведена таблиця Т1, що вже знаходиться у 2НФ, бо умова повної функціональної залежності виконується.

Для переходу до 3НФ необхідно позбавитися від транзитивних відношень в таблицях в 2НФ. Простіше кажучи, правило вимагає винести всі не ключові поля, вміст яких може не стосуватися записів таблиці в окремі таблиці. У Т1 є наступні транзитивні відношення: від Id власника клубу залежать ім’я, вік, адреса, рахунок; від Id порту залежать назва та місце розташування; від Id яхт-клубу залежать: назва, рахунок; від Id власника яхти залежить: ім’я, вік, адреса, рахунок. Позбавившись у Т1 від транзитивних залежностей, ключовими атрибутами будуть: Id власника клубу, Id порту, Id яхт-клубу, Id власника яхти. Спочатку виділимо з Т1 Id власника яхти (див. рис. 3.4).

Власник яхт-клубу (3НФ)

Id власника клубу

Ім’я

Вік

Адреса

Рахунок

Рисунок 3.4 – таблиця «Власник яхт-клубу»

Т2 (2НФ)

\*Id яхти

Назва

Тип

Рік випуску

Висота

Висота до палуби

Ширина

Довжина

Ціна

Id Власника яхти

Ім’я

Вік

Адреса

Рахунок

Id яхт-клубу

Назва

Рахунок

Id порту

Місце розташування

Назва

Id власника клубу

Рисунок 3.5 – таблиця Т2 у 2НФ після винесення власника яхт-клубу

Тепер виділимо з Т2 Id порту та отримаємо таблицю Т3 у 2НФ та таблицю Порт у 3НФ (див. рис. 3.6, 3.7).

Порт (3НФ)

Id порту

Місце розташування

Назва

Рисунок 3.6 – таблиця «Порт»

Т3 (2НФ)

\*Id яхти

Назва

Тип

Рік випуску

Висота

Висота до палуби

Ширина

Довжина

Ціна

Id Власника яхти

Ім’я

Вік

Адреса

Рахунок

Id яхт-клубу

Назва

Рахунок

Id порту

Id власника клубу

Рисунок 3.7 – таблиця Т3 у 2НФ після винесення порту

Виділимо з Т3 Id порту та отримаємо таблицю Т4 у 2НФ та таблицю Яхт-клуб у 3НФ (див. рис. 3.7, 3.8).

Яхт-клуб (3НФ)

Id яхт-клубу

Назва

Рахунок

Id порту

Id власника клубу

Рисунок 3.8 – таблиця «Яхт-клуб»

Т4 (2НФ)

\*Id яхти

Назва

Тип

Рік випуску

Висота

Висота до палуби

Ширина

Довжина

Ціна

Id Власника яхти

Ім’я

Вік

Адреса

Рахунок

Id яхт-клубу

Рисунок 3.9 – таблиця Т4 у 2НФ після винесення яхт-клубу

Виділимо з Т4 Id власника яхти та отримаємо таблиці Власник яхти та Яхти у 3НФ (див. рис. 3.10, 3.11).

Власник яхти(3НФ)

Id Власника яхти

Ім’я

Вік

Адреса

Рахунок

Id яхт-клубу

Рисунок 3.10 – таблиця «Власник яхти»

Яхта(3НФ)

\*Id яхти

Назва

Тип

Рік випуску

Висота

Висота до палуби

Ширина

Довжина

Ціна

Id Власника яхти

Id порту

Рисунок 3.11 – таблиця «Яхта»

В результаті нормалізації була отримана наступна схема бази даних (Див. Рис. 3.12)

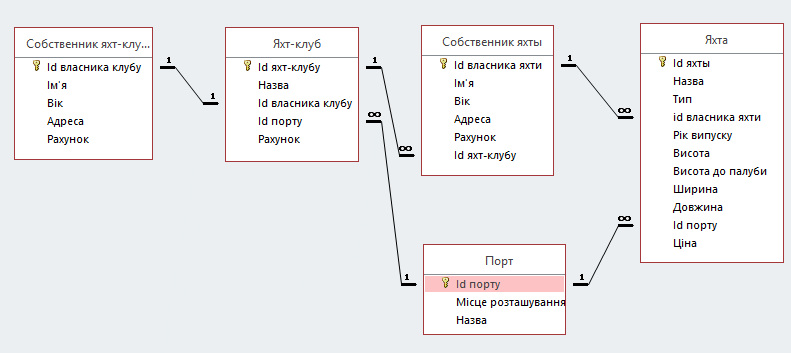


Рисунок 3.12 – кінцева схема даних

**4 ОПИС ПРОГРАМИ**

**4.1 Загальні відомості**

Програмна система «Яхт-клуб» – це десктопна програма, що створювався за допомогою середи розробки Visual Studio 2015. Розробка велася за допомогою мови програмування С# (серверна частина) та використовувалися Windows forms для візуалізації проекту. В якості СУБД був обраний Microsoft SQL Server, вбудований в середу розробки Visual Studio 2015. SQL Server Express LocalDB використовується для доступу до файлу бази даних SQL Server формату mdf.

Програма створена за допомогою шаблона проектування MVC (Model View Controller), що дозволяє поділити програму на три рівні.

Програма працюватиме на всіх комп’ютерах, для її функціонування не потрібне додаткове програмне забезпечення.

Розмір програм – 95Мб.

4.2 Виклик і завантаження

Після завантаження програми на ПК, потрібно підключити базу даних на ПК. Для цього потрібно відкрити файл App.config за допомогою будь-якого редактора та змінити шлях до бази даних (Yacht-club.mdf, що знаходиться в одній папці з App.config) у рядку ConnectionString. Після цього потрібно відкрити Сonstr.cs та повторити дії з ConnectionString.

Тепер при запуску програми та внесенні змін до БД, зміни будуть зберігатися у вас на комп’ютері.

4.3 Призначення і логічна структура

Програмна система виконує наступні функції:

1. додавання, зберігання, редагування, видалення, перегляд інформації про кожний об’єкт/сутність в базі даних;
2. пошук, сортування, фільтрація інформації про кожний об’єкт/сутність в базі даних;

Пошук в програмі за замовчуванням здійснюється по всім атрибутам об’єкта, але користувач може виконати його за обраним полем вибравши розширений пошук. Пошук відрізняється від фільтрації тим, що непідходящі записи не відкидаються, а підходящі підсвічуються жовтим.

Сортування можна виконувати за будь-яким одним атрибутом сутності за зростанням чи спаданням.

Фільтрація здійснюється з урахуванням всіх полів сутності залишаючи лише ті записи, які задовольняють параметри фільтрації.

1. формування статистики;

Статистика в програмі представлена у вигляді таблиць, що задаються за допомогою декількох атрибутів з різних таблиць бази даних.

1. завантаження звітів;

У зробити звіт користувач може спочатку переглянути дані, що будуть включені до цього звіту, далі обрати формат файлу для звіту. На даний момент звіт може бути згенеровано у PDF-формат або у HTML-формат.

1. виконання задачі автоматизації.

Користувач, ввівши тематику екскурсій, дату екскурсій, вид, кількість людей, місто та час, відведений на екскурсій, отримає всі можливі комбінації екскурсій по місцеположенням для цього обраного міста, загальний час яких не перевищує часу, введеного користувачем. Також є можливість додати певну комбінацію екскурсій до бази даних.   
Користувач, ввівши свій унікальний ідентифікатор, може обрати свою яхту та місце, куди він збирається. Після заповнення всіх необхідних даних користувач програми зможе побачити погодні умови того регіону та інформації щодо безпечності подорожі.

4.4 Опис фізичної моделі бази даних

В якості бази даних для програмного продукту «Яхт клуб» використовується локальна база даних Local DB, що вбудовано до середи розробки Visual Studio. При створенні в Visual Studio бази даних, заснованої на службах, для доступу до файлу бази даних (MDF) вона використовує підсистему SQL Server Express LocalDB.

До переваги локальної бази даних можна віднести те, що при використанні локальних даних програма підключається до файлу бази даних на локальному комп'ютері, а не до бази даних на окремому сервері. При підключенні до локальних даних, ви можете не тільки підключитися до файлу бази даних, а й інтегрувати його в програму. Переваги SQL Server Express LocalDB у тому, що ця система сумісна з заснованими на службах випусками SQL Server для функцій, що включаються SQL Server Express LocalDB. У SQL Server можна перемістити всі бази, не виконуючи ніяких дій по оновленню. Тому можна використовувати SQL Server Express LocalDB для розробки програм, орієнтованих на всі випуски SQL Server.

Нижче наведені структури наступних таблиць в базі даних: Яхти, Яхт-клуби, Власники яхт-клубів, Власники яхт, Порти, Метеостанція (див. рис. 4.1 – 4.6).

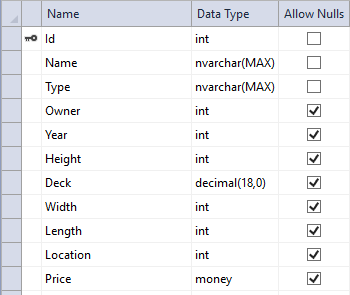


Рисунок 4.1 - Структура таблиці Яхти

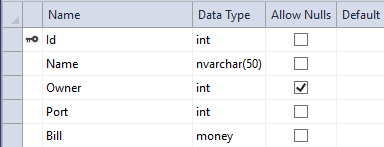


Рисунок 4.2 – Структура таблиці Яхт-клуби

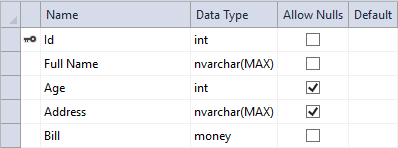


Рисунок 4.3 - Структура таблиці Власники яхт-клубів

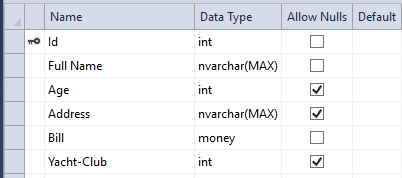


Рисунок 4.4 - Структура таблиці Власники яхт

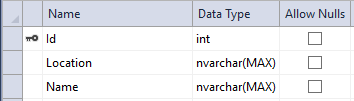


Рисунок 4.5 - Структура таблиці Порти

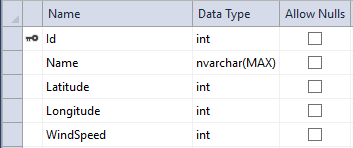


Рисунок 4.6 - Структура таблиці Метеостанція

4.5 Опис програмної реалізації

Додавання, редагування, видалення реалізовано на одному з головних вікон програми. Ці опції доступні для будь-якої таблиці, маніпулювання віж якими здійснюється за допомогою стрілок у лівій частині вікна. (див. рис. 4.9). При натисненні на додавання або редагування з'явиться нове вікно.( див. рис. 4.10).

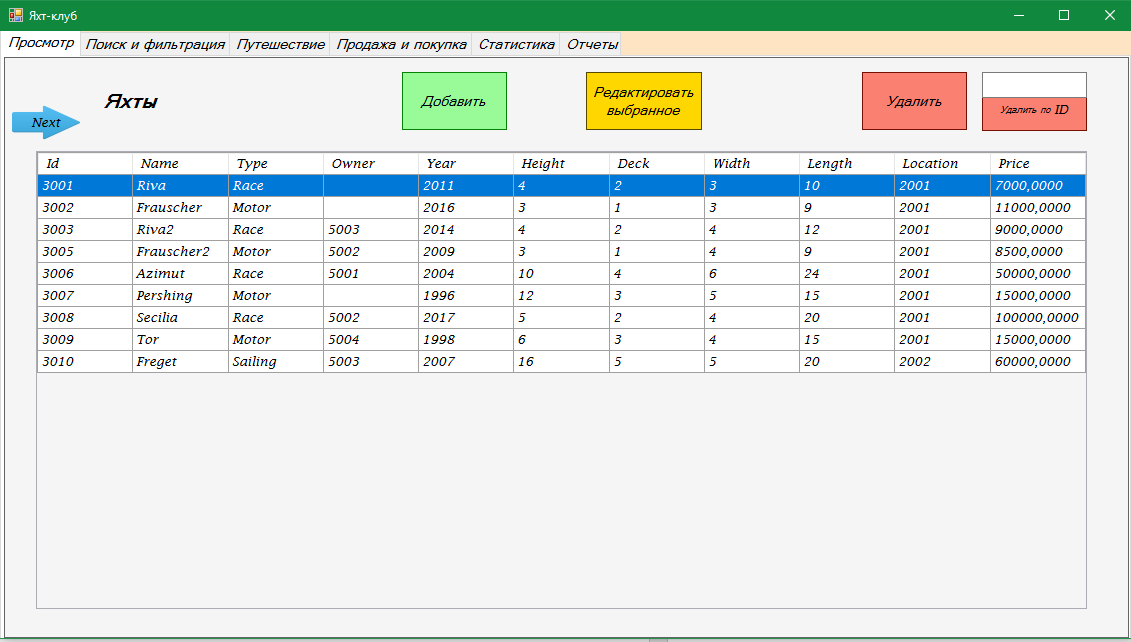


Рисунок 4.9 – Сторінка редагування даних про скульптуру

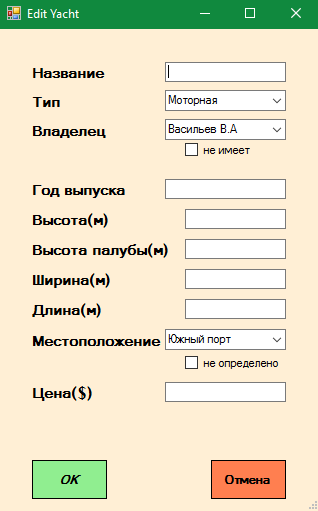


Рисунок 4.10 – Форма додавання/редагування даних про яхту.

При натисканні на видалення користувач повинен підтвердити дію ( актуально у разі випадкового натискання).(див. рис. 4.11).

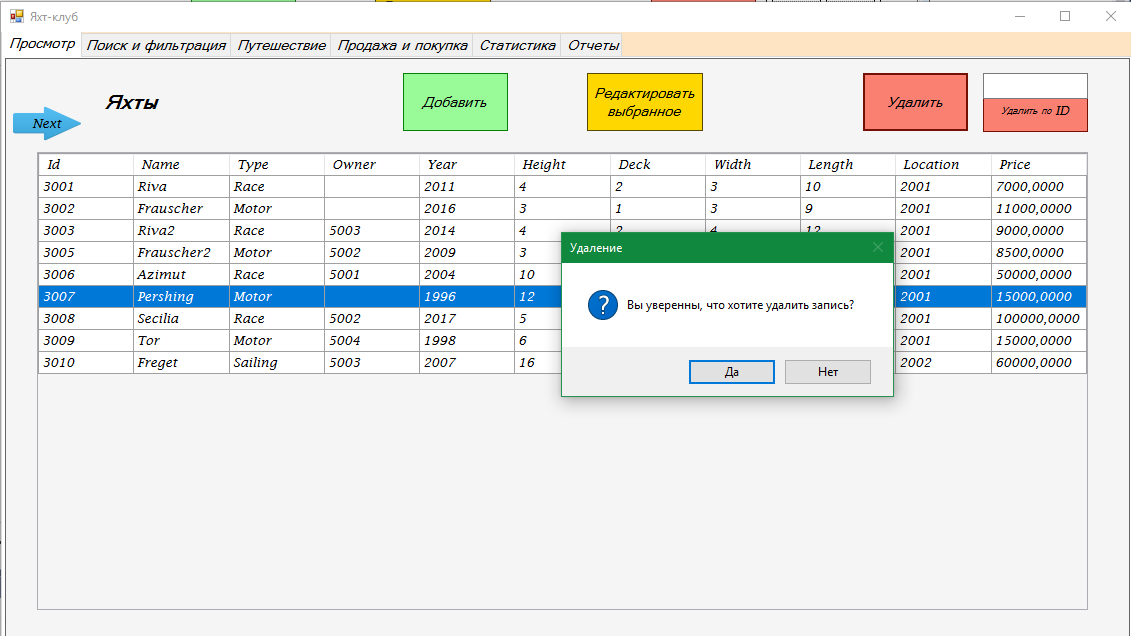


Рисунок 4.11 – Вікно підтвердження про видалення яхти

Пошук, фільтрація та сортування інформації знаходяться на сторінці пошуку та фільтрації меню програми (див. рис. 4.12).

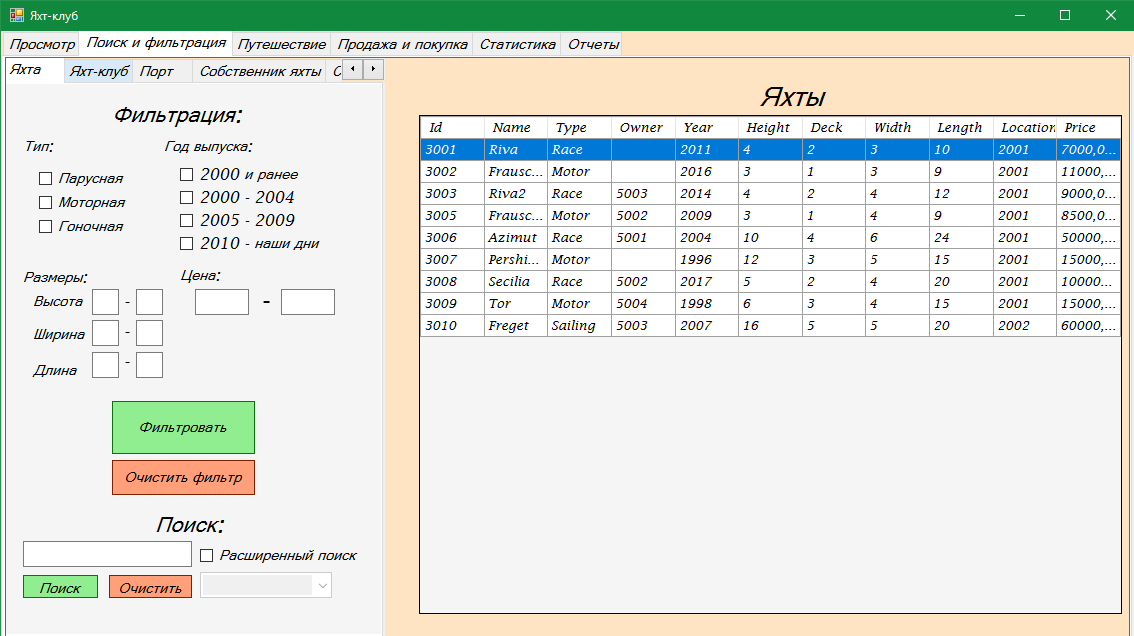


Рисунок 4.12 – Пошук, фільтрація та сортуванн

Для перегляду статистику потрібно в меню програми натиснути на вкладку «Статистика» та обрати потрібну статистику (див. рис. 4.13). Приклад статистики наведений на рисунку. 4.13.

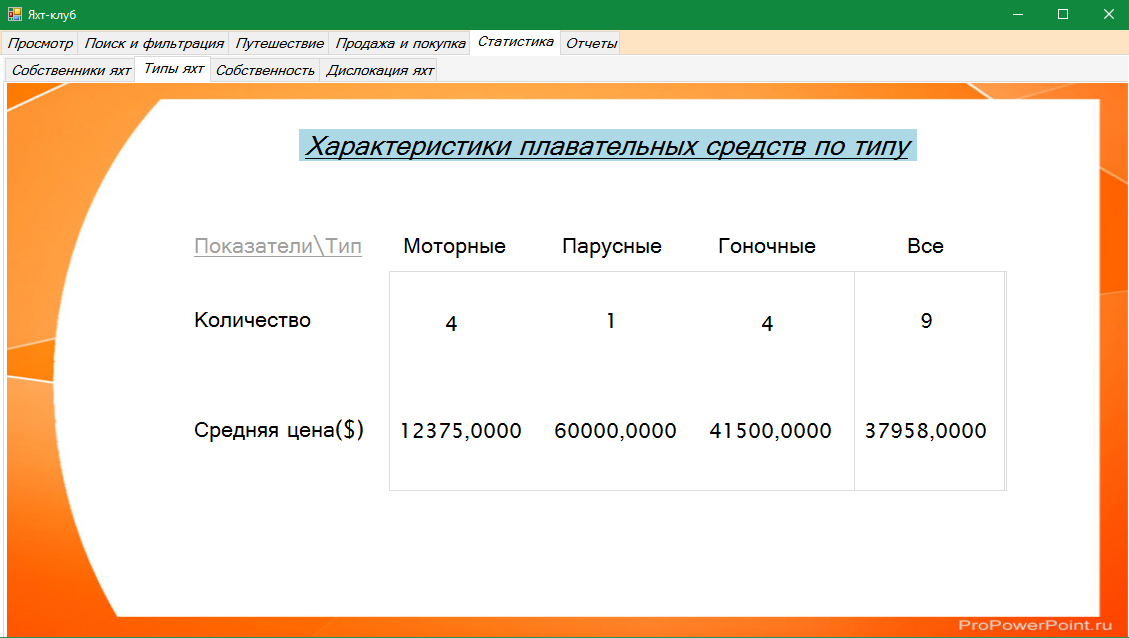


Рисунок 4.14 – Приклад статистики

Для перегляду звітів потрібно в меню програми натиснути на вкладку «Отчеты» та обрати потрібний звіт (див. рис. 4.15). На рисунку зображено звіт для власників яхт-клубів ,що надає інформацію про яхт-клуб після вводу номеру власника яхт-клубу. Користувач матиме можливість переглянути зміст звіту та обрати варіант формування звіту. На рисунку 4.17 – приклад звіту.

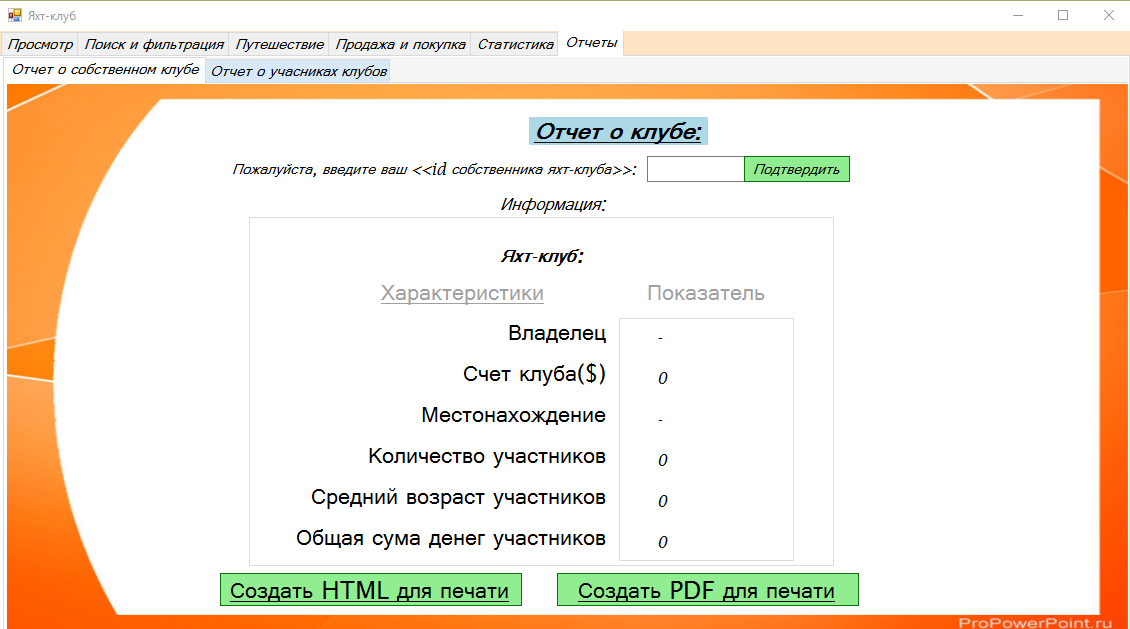


Рисунок 4.15 – Панель звітів

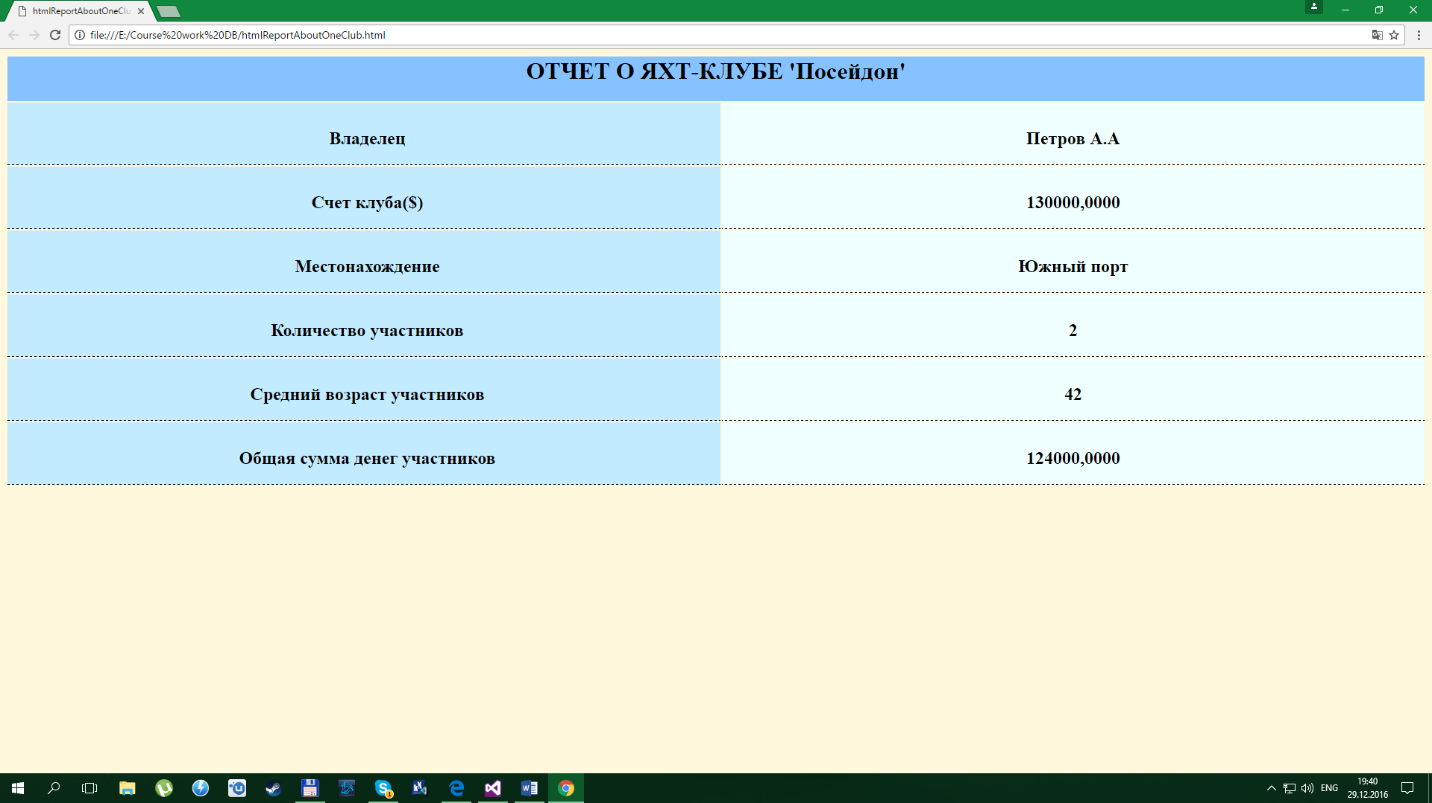


Рисунок 4.17 – Приклад звіту у HTML форматі

Система виконує такі види запитів для статистики:

1. статистика про кількість яхт, для кожного з типів ( моторна, парусна, гоночна);

SELECT COUNT(\*) FROM Yacht WHERE Type = 'motor';

SELECT AVG(Price) FROM Yacht WHERE Type = 'motor';

SELECT COUNT(\*) FROM Yacht WHERE Type = 'sailing';

SELECT AVG(Price) FROM Yacht WHERE Type = 'sailing';

SELECT COUNT(\*) FROM Yacht WHERE Type = 'race';

SELECT AVG(Price) FROM Yacht WHERE Type = 'race';

1. статистика про власників яхт

SELECT AVG(Age) FROM [Yacht owner];

SELECT AVG(Bill) FROM [Yacht owner];

SELECT SUM(Bill) FROM [Yacht owner];

SELECT Name FROM [Yacht-club] WHERE Id = (SELECT [Yacht-Club] FROM [Yacht owner] GROUP BY [Yacht-Club] HAVING COUNT(Id) >= ALL(SELECT COUNT(Id) FROM [Yacht owner] GROUP BY [Yacht-club]));

1. статистика о яхтах по виду собственности

SELECT COUNT(\*) FROM Yacht WHERE Owner IS NOT NULL;

SELECT COUNT(\*) FROM Yacht WHERE Owner IS NULL;

SELECT AVG(Price) FROM Yacht WHERE Owner IS NOT NULL;

SELECT AVG(Price) FROM Yacht WHERE Owner IS NULL;

1. статистика о яхтах по варинатах рамещения ( в порту, в аольном плаваньи)

SELECT COUNT(\*) FROM Yacht WHERE Location IS NULL;

SELECT COUNT(\*) FROM Yacht WHERE Location IS NOT NULL;

SELECT SUM(Price) FROM Yacht WHERE Location IS NOT NULL;

SELECT SUM(Price) FROM Yacht WHERE Location IS NULL;

Для експлуатації задачі автоматизації необхідно в меню програми обрати вкладку «Путешествие», ввести потрібні дані та отримати результат. Більш детально задача автоматизації, код та її алгоритм описані в розділі 4.6.

4.6 Опис задачі автоматизації

У випадку, коли власник яхти планує подорож куди-небудь, електронний менеджер подорожей може йому в цьому допомогти (див. рис. 4.18). Він повинен ввести свої дані, обрати транспорт на якому буду здійснюватися подорож, обрати пункт призначення. У вікні що з'явилося поруч користувач побачить характеристики яхти на якій він збирається у подорож, місце подорожі, а також погодні умови на заданій території (див. рис. 4.19).

Система отримує з бази даних інформацію щодо швидкості вітру на території та вираховує висоту хвилі. У подальшому характеристики яхти користувача за деякими законами та формулами зіставляються з висотою хвиль. Користувач отримає інформацію щодо безпечності подорожі на даний момент.

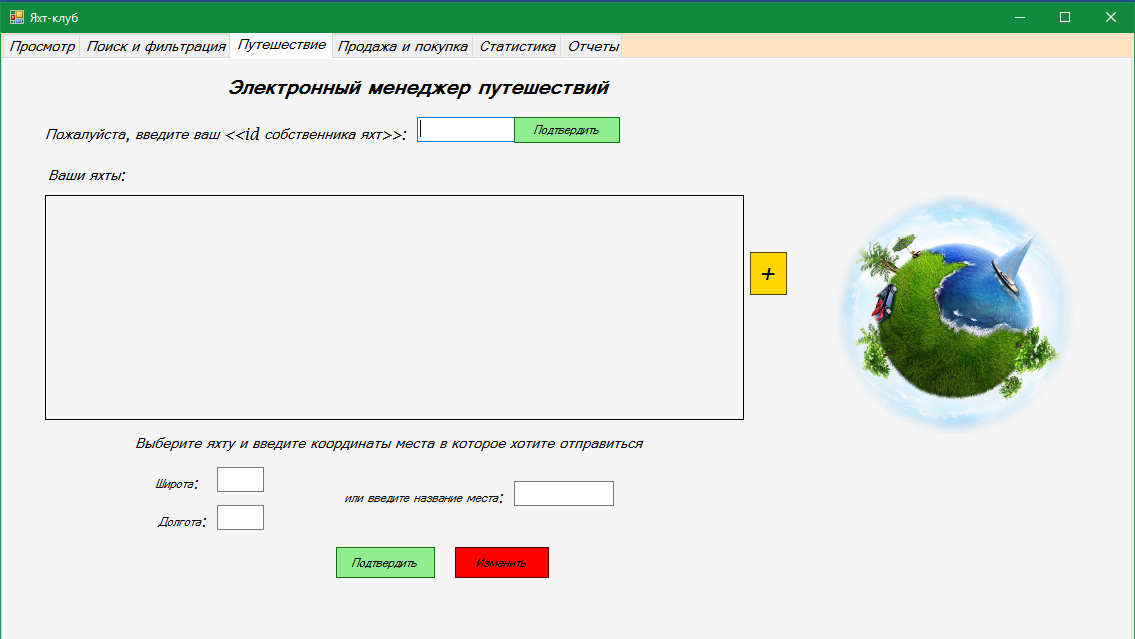


Рисунок 4.18 – Вікно менеджера подорожей

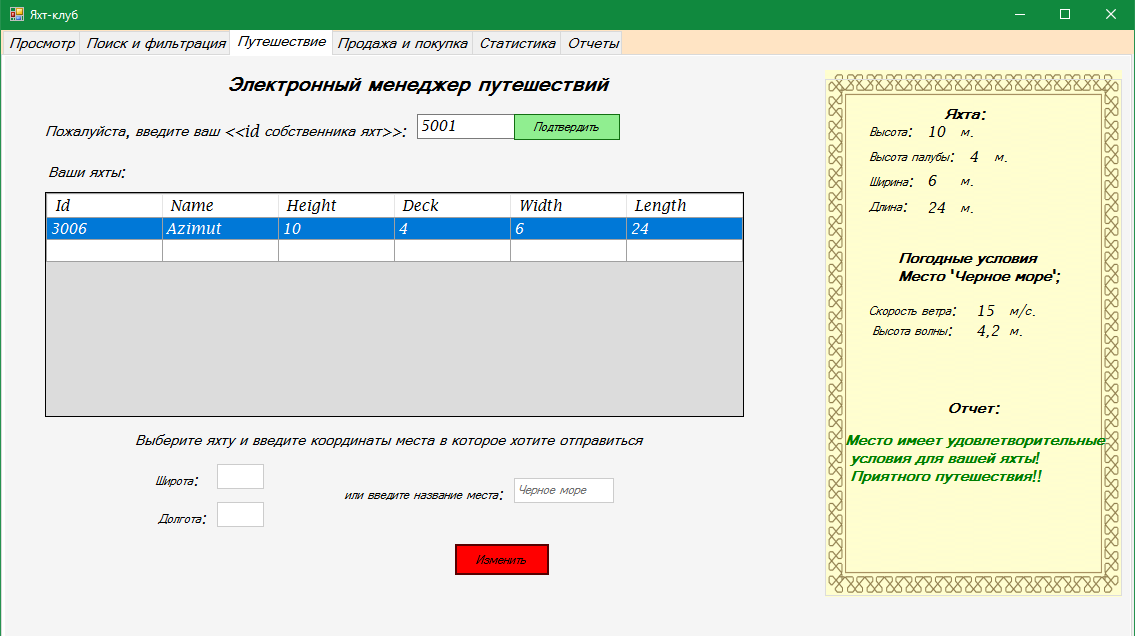


Рисунок 4.19 – Вікно інформації менеджера подорожей

Алгоритм задачі автоматизації наступний:

1. система зчитує дані введені користувачем про його яхту;
2. система зчитує дані про пункт призначення подорожі;
3. система за допомогою спеціального алгоритму ( наведено нижче) виконує розрхунки з метою визначення того факту, чи безпечні умови для яхти у тому регіоні;
4. система виводить у результаті усі введені дані, інформацію про погодні умови у вибраному регіоні та повідомлення про безпечність чи небезпечність подорожі .

Код алгоритму задачі автоматизації:

//ЗАДАЧА АВТОМАТИЗАЦИИ

public string height;

public string deck;

public string width;

public string length;

public string longitude;

public string latitude;

public string name;

//выбор пользователя при автоматизации

private void buttonAcceptId\_Click(object sender, EventArgs e)

{

textBoxOwnersId.BackColor = Color.White;

if (textBoxOwnersId.Text == "")

{

textBoxOwnersId.BackColor = Color.LightSalmon;

return;

}

string strFirstCase = "SELECT Id, Name, Height, Deck, Width, Length FROM Yacht WHERE Owner = ";

strFirstCase += textBoxOwnersId.Text;

SqlConnection sqlconn = new SqlConnection(Constring);

sqlconn.Open();

SqlDataAdapter oda = new SqlDataAdapter(strFirstCase, sqlconn);

DataTable dt = new DataTable();

oda.Fill(dt);

dataGridView3.DataSource = dt;

sqlconn.Close();

}

//выбор яхты при автоматизации

private void button6\_Click(object sender, EventArgs e)

{

textBoxOwnersId.BackColor = Color.White;

if (textBoxOwnersId.Text == "")

{

textBoxOwnersId.BackColor = Color.LightSalmon;

return;

}

height = dataGridView3.SelectedRows[0].Cells[2].Value.ToString();

deck = dataGridView3.SelectedRows[0].Cells[3].Value.ToString();

width = dataGridView3.SelectedRows[0].Cells[4].Value.ToString();

length = dataGridView3.SelectedRows[0].Cells[5].Value.ToString();

dataGridView3.Enabled = false;

dataGridView3.BackgroundColor = Color.Gainsboro;

button6.Visible = false;

buttonChangeYacht.Visible = true;

}

//изменение яхты при автоматизации

private void buttonChangeYacht\_Click(object sender, EventArgs e)

{

button6.Visible = true;

buttonChangeYacht.Visible = false;

dataGridView3.Enabled = true;

dataGridView3.BackgroundColor = Color.WhiteSmoke;

}

//автоматизация реализация

private void button5\_Click(object sender, EventArgs e)

{

textBoxOwnersId.BackColor = Color.White;

if (textBoxOwnersId.Text == "")

{

textBoxOwnersId.BackColor = Color.LightSalmon;

return;

}

textBoxLatitude.BackColor = Color.White;

textBoxLongitude.BackColor = Color.White;

textBoxNamePlace.BackColor = Color.White;

if((textBoxLatitude.Text == "" || textBoxLongitude.Text == "") && textBoxNamePlace.Text == "")

{

textBoxLatitude.BackColor = Color.LightSalmon;

textBoxLongitude.BackColor = Color.LightSalmon;

textBoxNamePlace.BackColor = Color.LightSalmon;

return;

}

groupBox1.Visible = true;

button5.Visible = false;

button7.Visible = true;

textBoxLatitude.Enabled = false;

textBoxLongitude.Enabled = false;

textBoxNamePlace.Enabled = false;

buttonChangeYacht.Visible = false;

latitude = textBoxLatitude.Text;

longitude = textBoxLongitude.Text;

name = textBoxNamePlace.Text;

label26.Text = height;

label27.Text = width;

label28.Text = length;

label50.Text = deck;

if(textBoxLatitude.Text != "" || textBoxLongitude.Text != "")

{

label44.Text = latitude;

label46.Text = longitude;

label46.Visible = true;

label44.Visible = true;

label47.Visible = true;

label48.Visible = true;

}

string nameTmp = "Место: " + name;

if(textBoxNamePlace.Text != "")

{

labelName.Text = nameTmp;

labelName.Visible = true;

}

if (latitude == "" && longitude == "")

{

labelName.Visible = true;

}

label40.Text = "";

//попытка достать место из БД

SqlConnection sqlconn = new SqlConnection(Constring);

sqlconn.Open();

string windDataBase = "";

if ((textBoxLatitude.Text == "" || textBoxLongitude.Text == "") && textBoxNamePlace.Text != "")

{

windDataBase = meteostationTableAdapter1.ScalarQueryWind(textBoxNamePlace.Text).ToString();

}

if((textBoxLatitude.Text != "" && textBoxLongitude.Text != "") && textBoxNamePlace.Text == "")

{

try

{

windDataBase = meteostationTableAdapter1.ScalarQueryWindXY(Convert.ToInt32(textBoxLatitude.Text), Convert.ToInt32(textBoxLongitude.Text)).ToString();

}

catch { }

}

label40.Text = windDataBase;

string wind = label40.Text;

if (label40.Text == "")

{

Random rand = new Random();

wind = rand.Next(30).ToString();

label40.Text = wind;

}

bool warning;

double wave = Methods.GetWave(wind, height, deck, out warning);

label39.Text = wave.ToString();

if (warning)

{

label42.Text = "Осторожно, погодные условия в \n данном регионе опасны для \n вашего судна. \n Рекомендуем воздержаться \n от этого путешествия!!";

label42.ForeColor = Color.Red;

}

else

{

label42.Text = "Место имеет удовлетворительные \n условия для вашей яхты! \n Приятного путешествия!!";

label42.ForeColor = Color.Green;

}

}

//измениние данных о путешествии

private void button7\_Click(object sender, EventArgs e)

{

textBoxOwnersId.BackColor = Color.White;

if (textBoxOwnersId.Text == "")

{

textBoxOwnersId.BackColor = Color.LightSalmon;

return;

}

label46.Visible = false;

label44.Visible = false;

label47.Visible = false;

label48.Visible = false;

labelName.Visible = false;

groupBox1.Visible = false;

button5.Visible = true;

button7.Visible = false;

textBoxLatitude.Enabled = true;

textBoxLongitude.Enabled = true;

textBoxNamePlace.Enabled = true;

buttonChangeYacht.Visible = true;

}

//метод розрахунку висоти хвилі

public static double GetWave(string wind,string height, string deck, out bool warning)

{

double wave = 0;

double i = 0;

warning = false;

if (Convert.ToInt32(wind) > 0 && Convert.ToInt32(wind) < 6)

{

i = 1.1;

}

if (Convert.ToInt32(wind) > 5 && Convert.ToInt32(wind) < 11)

{

i = 1.2;

}

if (Convert.ToInt32(wind) > 10 && Convert.ToInt32(wind) < 16)

{

i = 1.4;

}

if (Convert.ToInt32(wind) > 15 && Convert.ToInt32(wind) < 21)

{

i = 1.75;

}

if (Convert.ToInt32(wind) > 20)

{

i = 2.2;

}

wave = (Convert.ToInt32(wind) \* 0.2) \* i;

if (Convert.ToInt32(height) < 30)

{

if (wave > (3 / 4) \* Convert.ToInt32(deck))

{

if (((Convert.ToInt32(height) - Convert.ToInt32(deck)) > (Convert.ToInt32(deck) \* 2)) || wave > Convert.ToInt32(height) + 0.5)

{

warning = true;

}

}

}

return wave;

}

5 ВИСНОВКИ

В результаті виконаної роботи була створена інформаційна система «Яхт-клуб». Система була розроблена як десктопна програма. Для реалізації потрібного функціоналу була проаналізована предметна область, описані взаємозв’язки між об’єктами інформаційної системи. Для зберігання інформації була спроектована та нормалізована база даних, яка відображає всі зв’язки між сутностями даної предметної області. Дана система повністю відповідає потребам користувачів, оскілька вона є швидким та ефективним засобом роботи, що значно зменшую часові затрати при роботі з програмою.

Заздалегідь продуманий зрозумілий інтерфейс дозволяє ефективно працювати з програмою, швидко знаходити всі її функції. Саме тому програма може одразу використовуватися людьми без спецальних знань.

Отримана система задовольняє заявленим до неї вимогам, в ній реалізовані пошук, фільтрація, сортування, задача автоматизації, а також можливість перегляду та формування статистики, завантаження звітів.

6 ПЕРЕЛІК ПОСИЛАНЬ

1. Герберт Шилдт - C# 4.0. Полное руководство: Пер. з англ. – Москва. : ООО «И.Д. Вільямс», 2011. – 1056с.

2. Бейли Л. Head First SQL. .: Ид-во “ Питер”, 2012. - 573с.

3. Мазурова О.О, Черепанова Ю.С., Широкопєтлева М.С. Методичні вказівки до курсової роботи з дисциплини “Бази даних”

4. Троелсен: «Мова програмування С# 2005 та платформа .NET 2.0» [Текст]: учеб. пособ, 3е видання. : Пер. з англ. – Москва. : ООО «И.Д. Вільямс», 2007. – 1168с.

5. Роберт Дж. Мюллер: «Database Design for Smarties. Using UML for Data Modeling» Ид-во “Лори”, 2002. – 420.