Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

ФКСиС

Кафедра ЭВМ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

«ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ОРГАНИЗАЦИИ БЭБИСИТЕРСТВА»

Студент гр. 444601 О.А. Швед

Руководитель С. А. Кучук

МИНСК 2016

Содержание

[Введение 3](#_Toc402190757)

[1 Обзор литературы](#_Toc402190758) 4

[2 Структурное проектирование](#_Toc402190759) 7

[3 Функциональное проектирование](#_Toc402190761) 8

[4 Тестирование 1](#_Toc402190761)0

[5 Руководство пользователя 1](#_Toc402190762)2

[Заключение 1](#_Toc402190763)6

[Список используемых источников 1](#_Toc402190764)7

**вВедение**

В данной курсовой работе будет разработано программное средство для организации бэбиситерства. Пользователь получит возможность вести учет нянь и клиентов. Без всякого труда сможет добавлять или удалять записи о них, также он может сохранять все данные в базе данных.

Бэбиситтерство- это практика временного ухода за ребёнком распространённая в США и странах Западной Европы. Обычно в качестве бэбиситтеров нанимают подростков или студентов, как правило девушек. В обязанности бэбиситтеров могут включать не только наблюдение за маленькими детьми, но и игры с ними, чтение им книг, обучение девочек кулинарии или даже обучение вождению автомобиля. В Республике Беларусь рынок бебиситтеров только зарождается, а в Европе это развитая сфера услуг, которой воспользовалось множество семей.

Планируется реализовать следующие функции:

- регистрация, изменение, удаление профиля няни (педагогический стаж, ВО, возраст, фото, привычки);

- возможность составлять расписание для няни;

- отчет «Расписание няни на неделю»;

- возможность выбрать вместе с пользователем по телефону няню на заданный интервал времени из всех, которые не заняты в него;

- возможность регистрации отпуска/дней по болезни нянь.

# **1 ОБЗОР ЛИТЕРАТУРЫ**

Согласно теме курсового проекта, задачей проектирования является разработка прикладного приложения.

Для выбора СУБД произведем анализ бесплатных популярных систем:

**MongoDB** — это документо-ориентированная СУБД. Данные в MongoDB хранятся в документах, которые объединяются в коллекции. Каждый документ представляет собой JSON-подобную структуру. Проведя аналогию с реляционными СУБД, можно сказать, что коллекциям соответствуют таблицы, а документам — строки в таблицах. Максимальный размер документа в MongoDB 2.x составляет 16 Мб (в более ранних версиях — лишь 4 Мб);В отличие от РСУБД MongoDB не требует какого-либо описания схемы базы данных — она может постепенно меняться по мере развития приложения, что есть удобно. Поддерживаются индексы, в том числе по массивам и вложенным документам, а также геопространственные индексы. Поддерживаются уникальные и составные индексы. Также MongoDB есть атомарные операции, compare-and-swap, [курсоры](http://www.mongodb.org/display/DOCS/Queries+and+Cursors), запись без подтверждения и даже MapReduce.Размер коллекции в MongoDB может быть ограничен числом документов или мегабайтами. Если коллекция слишком разрастется, старые документы будут удалены. Эта возможность может пригодится, если вы собираетесь хранить в MongoDB какие-то временные данные. Интерфейс MongoDB сильно напоминают работу с [DBIx::Class](http://eax.me/dbix-class/). В запросах могут использоваться функции на языке JavaScript.В MongoDB поддерживается журналирование, а также асинхронная репликация двух видов — [master-slave репликация](http://eax.me/postgresql-replication/) и наборы реплик. Набор реплик представляет собой тот же master-slave, но в случае падения мастера среди реплик автоматически выбирается новый мастер. После возобновления своей работы бывший мастер становится репликой. Пожалуй, самая значительная особенность MongoDB заключается в том, что документы могут быть автоматически сегментированы по нескольким наборам реплик.

**MySQL** – вторая из самых используемая в мире open-source реляционная СУБД. Данная СУБД используется во многих крупных программных продуктах и на многих веб-сайтах, в том числе YouTube, Wikipedia, Facebook, и многих других. Система полностью бесплатна (в том числе и для коммерческого использования).MySQL создан с помощью языков C/C++, однако большое количество языков программирования включают в себя APIдля работы с MySQL (MySQLConnector / Netдля C# иVB, JDBC driver для Java, MyODBC для языков, поддерживающих интерфейс ODBC).

**Microsoft SQL Server** — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

**PostgreSQL**– мощная и тяжелая система, отвечающая всем современным стандартам СУБД. Больше подходит для серьезных проектов, требующих сложных баз данных. По скорости работы **PostgreSQL** уступает MySQL. Также данная СУБД сложнее в администрировании, но является не чисто реляционной, а реляционно-объектной, что позволяет ей легко отражать иерархии объектов на реляционные таблицы.

* 1. **Обоснование выбора СУБД**

В качестве СУБД будет использоваться MSQL, которая является надежной СУБД, не требующей высокопроизводительных компьютеров.

**MSQL Server** является надежной базой данных для любых целей, может продолжать расширяться по мере наполнения информацией, без заметного уменьшения быстродействия операций с записями в многопользовательском режиме. Пользователи могут быть добавлены путем модернизации оборудования. В последнем тесте поддерживалось до 4600 пользователей базы данных.   
 Обеспечивается максимальная безопасность.  Ваши данные защищены от несанкционированного доступа за счет интеграции сетевой безопасности с сервером безопасности.   Поскольку безопасность на уровне пользователя, пользователи могут иметь ограниченный доступ к записи данных, тем самым защищая их от модификации или поиска, указав доступ на уровне пользовательских привилегией.  Кроме того, с данными, хранящимися на отдельном сервере, сервер работает как шлюз, который ограничивает несанкционированный доступ.

MSQL Server обрабатывает запросы от пользователей и только отправляет пользователю результаты запроса.   Таким образом, минимальная информация передается по сети.   Это улучшает время отклика и устраняет узкие места в сети.   Это также позволяет использовать MSQL Server в качестве идеальной базы данных для интернет .

Техническое обслуживание  MSQL Server очень простое и не требует больших знаний. Возможны изменения в структуре данных а так же резервное копирование во время работы сервера, без остановки.

 SQL Server является приложением базы данных  при работе на   . Net, новейшие разработки Microsoft.   Выбрав Microsoft SQL Sever в качестве базы данных информации для компании, приложение может расширяться и адаптироваться по мере изменения бизнес-климата.

* 1. **Выбор GUIбиблиотеки. Анализ популярных продуктов**

Здесь будут рассмотрены : GIMP ToolKit, Qt, Windows Forms**.**

**GTK+** (сокращение от **GIMP ToolKit**) — кроссплатформенная библиотека элементов интерфейса, имеет простой в использовании API, наряду с Qt является одной из двух наиболее популярных на сегодняшний день библиотек для X Window System. Будучи изначально частью графического редактора GIMP, она развилась в отдельный проект и приобрела заметную популярность. GTK+ — свободное ПО, распространяемое на условиях GNU LGPL, позволяющей создавать как свободное, так и проприетарное программное обеспечение с использованием библиотеки. GTK+ является официальной библиотекой для создания графического интерфейса проекта GNU.

**Qt**— кроссплатформенный инструментарий разработки ПО на языке программирования, также имеющий привязки ко многим другим языкам программирования. Позволяет запускать написанное с его помощью ПО в большинстве современных операционных систем путём простой компиляции программы для каждой ОС без изменения исходного кода. Включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Qt является полностью объектно-ориентированным, легко расширяемым и поддерживающим технику компонентного программирования.

**Windows Forms** – элемент Фреймворка .NET, построенный на основе устаревшего MFC и широко используемый в разработке графических интерфейсов многих приложений. Windows Forms предоставляет возможность разработки кроссплатформенного графического пользовательского интерфейса.

В качестве GUIбиблиотеки данной работе будет использован Windows Forms, так как он прост в использовании и, в отличие от других библиотек, не требует установки дополнительных компонентов для разработки приложения, NETфреймворк версии 3.0 или выше установлен на большей части используемых сейчас машин.

**2 Структурное проектирование**

Проектируемое программное средство будет состоять из компонентов, изображенных на рисунке 2.1.

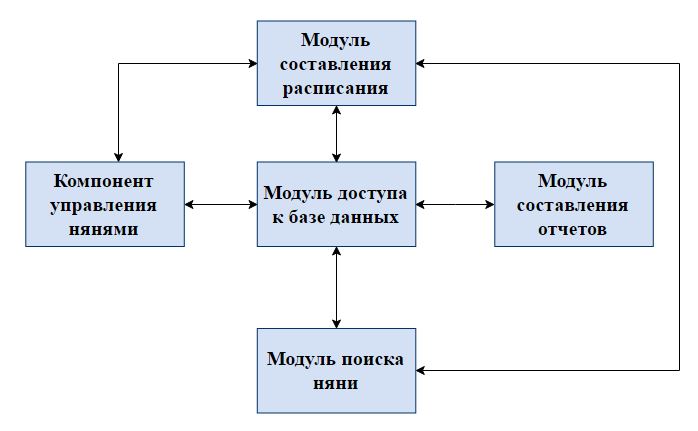


Рисунок 2.1 – Структурная схема программного средства

Описание компонентов, изображенных на рисунке 2.1:

* модуль доступа к БД – это компонент, предназначенный для получения информации из базы данных и записи информации в нее;
* модуль составления отчетов – это компонент, предназначенный для составления различных отчетов;
* модуль поиска няни – это компонент, предназначенный для поиска нянь по определенным критериям;
* компонент управления нянями – это компонент, предназначенный для управления профилями нянь;
* модуль составления расписания – это компонент, предназначенный для составления расписания для нянь.

База данных разрабатываемого приложения будет иметь структуру, показанную на рисунке 2.2.

Таблица «Няня» будет использоваться для хранения информации о нянях;

Таблица «Отпуск няни» будет использоваться для хранения информации об отпусках нянь;

Таблица «Расписание» будет использоваться для хранения данных о расписании для нянь;

Таблица «Заявки» будет использоваться для хранения информации о заявках на нянь;

Таблица «Клиент» будет использоваться для хранения информации о клиентах.

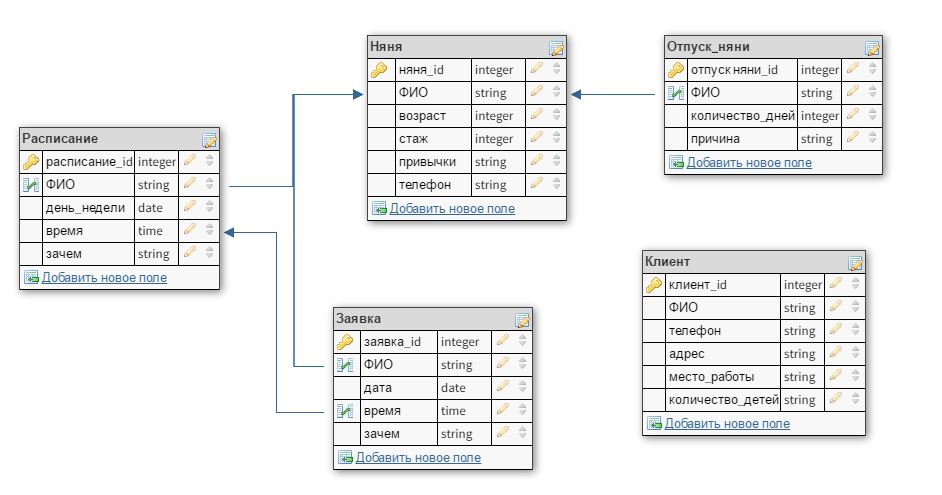


Рисунок 2.2 – Схема базы данных

Таким образом в ходе структурного проектирования были выделены основные компоненты приложения, а также спроектирована схема базы данных.

**3 Функциональное проектирование**

Проектируемое программное средство будет включать в себя следующие модули:

* модуль доступа к БД;
* модуль составления отчетов;
* модуль поиска няни;
* компонент управления нянями;
* модуль составления расписания.
  1. **Модуль доступа к базе данных**

Данный модуль предназначен для получения информации из базы данных и записи информации в нее. Модуль состоит из набора классов, представленных на рисунке 3.1.

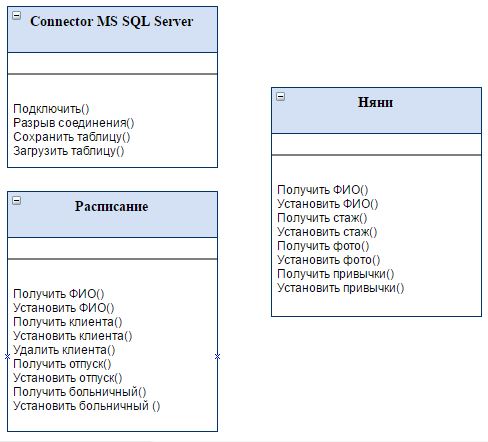


Рисунок 3.1 – Диаграмма классов модуля доступа к базе данных

Класс «Connector MS SQL Server» будет отвечать за подключение БД.

Класс «Няни» будет заниматься получением и записью информации о нянях в БД. Класс «Расписание» будет заниматься получением и записью информации о расписании нянь в БД.

* 1. **Модуль составления отчетов**

Данный модуль предназначен для составления различного рода отчётов и экспорта их в Exсel-файл. Модуль состоит из набора классов, представленных на рисунке 3.2.

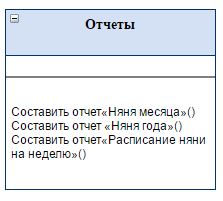


Рисунок 3.2 – Диаграмма классов модуля составления отчетов

Класс «Отчеты» будет заниматься составлением отчетов, который в данный момент нужен пользователю.

* 1. **Модуль поиска няни**

Данный модуль предназначен поиска няни по определенным критериям. Модуль состоит из набора классов, представленных на рисунке 3.3.

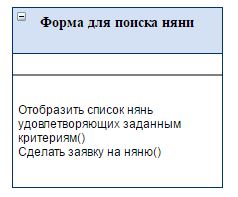


Рисунок 3.3 – Диаграмма классов модуля поиска няни

Класс «Форма для поиска няни» будет заниматься поиском нянь по заданным критериям, так же будет возможность сделать заявку на интересующую няню.

* 1. **Компонент управления нянями**

Данный компонент предназначен для регистрации, изменения, удаления профиля няни. Компонент состоит из набора классов, представленных на рисунке 3.4.

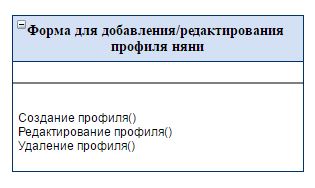


Рисунок 3.4 – Диаграмма классов компонента управления нянями

Класс «Форма для добавления/редактирования профиля няни» будет создавать профили новых нянь, а также редактировать или удалять уже существующие профили.

* 1. **Модуль составления расписания**

Данный компонент предназначен для составления расписания для нянь, а также для планирования отпусков и больничных. Модуль состоит из набора классов, представленных на рисунке 3.4.

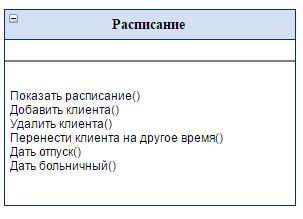


Рисунок 3.4 – Диаграмма классов модуля составления расписания

Класс «Расписание» будет заниматься составлением расписания для нянь, а также планировать отпуска и больничные.

**3 Тестирование**

Целью тестирования является проверка работоспособности разработанного программного обеспечения. На этой стадии необходимо проверить корректное функционирование разработанного программного обеспечения и соответствие его требованиям, выдвинутым в техническом задании. При выявлении несоответствий работы программы техническому заданию, либо ошибок, требуется доработка программного обеспечения и (или) документации. Разработанное программное обеспечение должно гарантировать устойчивое функционирование независимо от действий конечных пользователей. При возникновении отдельных ошибочных ситуаций разработанное программное обеспечение должно их успешно обрабатывать. Для проведения корректного тестирования сначала необходимо разработать порядок испытаний. Порядок испытаний - это список последовательности действий направленных на проверку корректности работы программы и (или) ее отдельных функциональных частей.

Существует два основных вида тестирования: функциональное и структурное. При функциональном тестировании программа рассматривается как “черный ящик” (то есть ее текст не используется). Происходит проверка соответствия поведения программы ее внешней спецификации. Возможно ли при этом полное тестирование программы? Очевидно, что критерием полноты тестирования в этом случае являлся бы перебор всех возможных значений входных данных, что невыполнимо.

Поскольку исчерпывающее функциональное тестирование невозможно, речь может идти о разработки методов, позволяющих подбирать тесты не “вслепую”, а с большой вероятностью обнаружения ошибок в программе. При структурном тестировании программа рассматривается как “белый ящик” (т.е. ее текст открыт для пользования). Происходит проверка логики программы. Полным тестированием в этом случае будет такое, которое приведет к перебору всех возможных путей на графе передач управления программы (ее управляющем графе). Даже для средних по сложности программ числом таких путей может достигать десятков тысяч.

Таким образом, ни структурное, ни функциональное тестирование не может быть исчерпывающим. Рассмотрим подробнее основные этапы тестирования программных комплексов. В тестирование многомодульных программных комплексов можно выделить четыре этапа:

Этапы тестирования приведены в таблице 4.1.

Таблица 4.1 – Этапы тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| Вид тестирования | Стадия, этап | Объект | Критерий |
| Структурное, надежности | Разработка | Компоненты | Покрытые ветвлений, функций |
| Сборочное | Разработка | Подсистемы | Функциональность, степень проверки компонентов |
| Функциональное | Разработка | Система в целом | Соответствие функциональным требованиям ТЗ |
| Регрессионное | Разработка, сопровождение | Система в целом | Проверка качества внесения изменений |
| Нагрузочное | Разработка, сопровождение | Система в целом | Оценка статистических характеристик системы, соответствие ТЗ, ТТХ, подбор конфигурация оборудования |
| Стрессовое | Разработка, сопровождение | Система в целом | Корректность работы системы при предельных нагрузках |

# **4** **Руководство пользователя**

Для работы компьютер должен удовлетворять минимальным системным требованиям.

Минимальные системные требования :

* Windows 7, Windows 8;
* RAM: 1024 Мб;
* HDD: 1 Гб;
* Устройства ввода: клавиатура, мышь.
* Установленный Microsoft SQL server 2012.

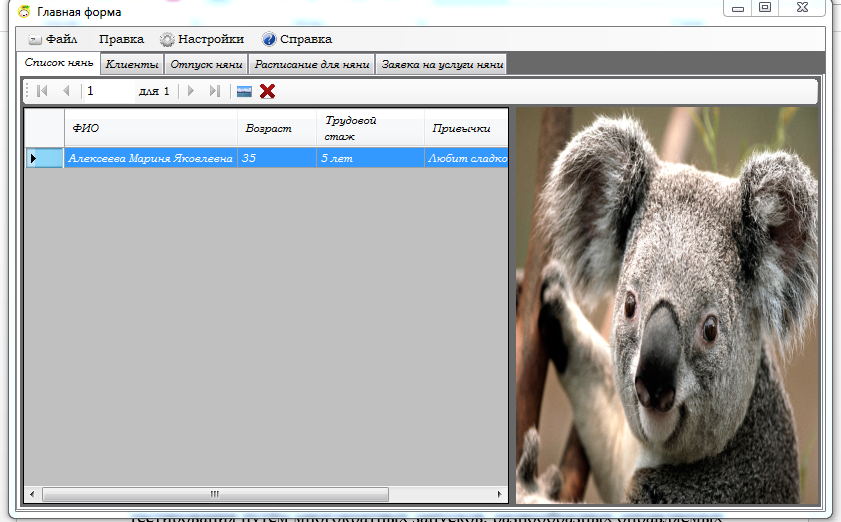


Рисунок 5.1 – Скриншот форма для работы с приложением

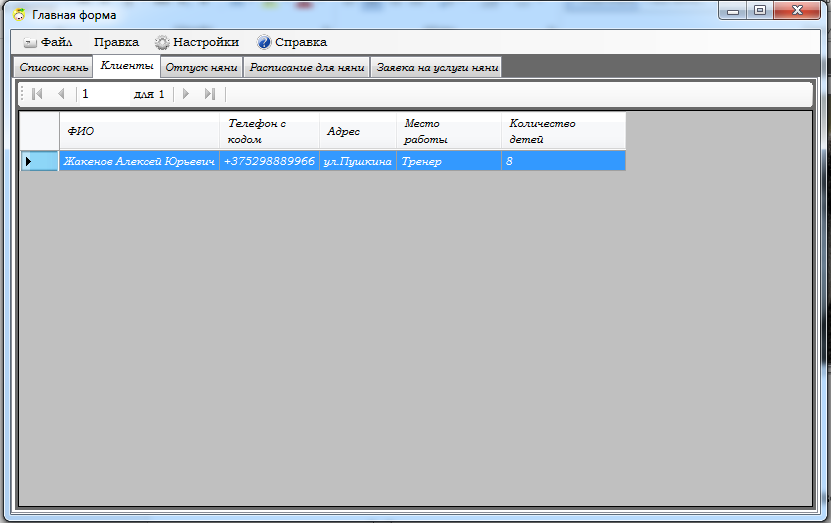


Рисунок 5.2 –Скриншот форма данных о клиентах

Для того чтобы добавить клиента, необходимо перейти на вкладку “Правка “выбрать пункт добавить, после этого ввести данные (см.рисунок 5.3).

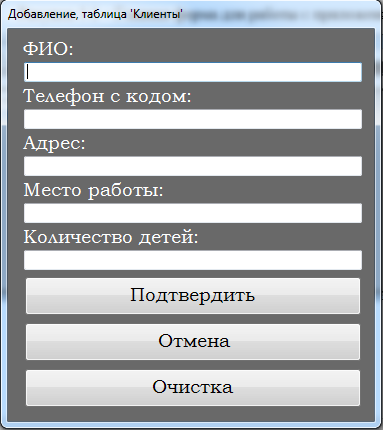


Рисунок 5.3 – Поля для заполнения данных о клиенте

Добавление данных в таблицу “Отпуск для няни” происходит ,как и добавление клиента, но заполняются другие данные, а так же причина ее ухода в отпуск(см.рисунок 5.4)

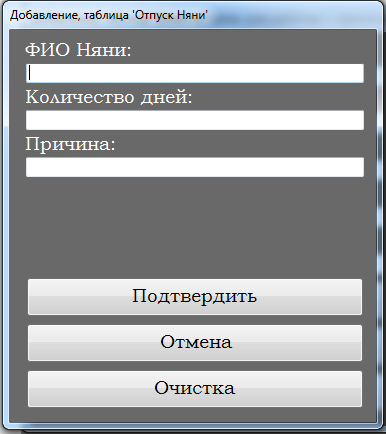


Рисунок 5.4- Уход няни в отпуск

Для того чтобы узнать, какие дела у няни запланированы на неделю, необходимо перейти на вкладку “Расписание няни”(см.рисунок 5.5)

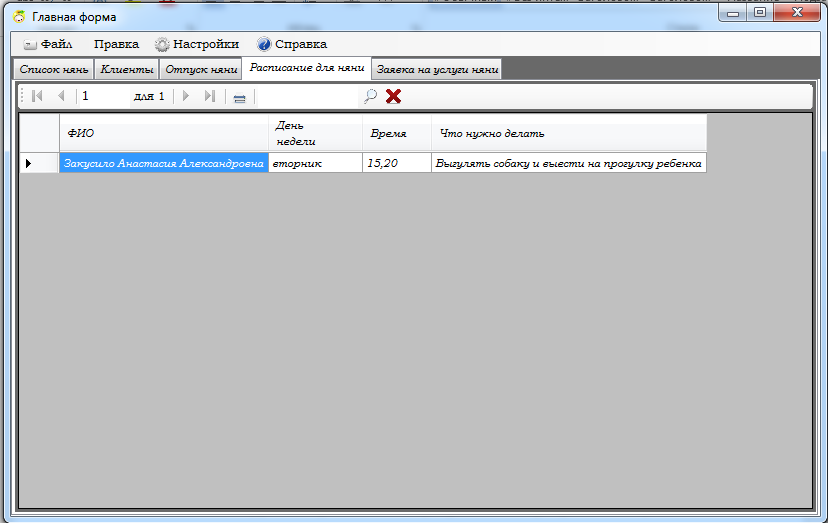


Рисунок 5.5 – Вкладка “Расписание для няни”

Для того чтобы добавить фото няни необходимо нажать на соответствующую иконку (см.рисунок 5.6).

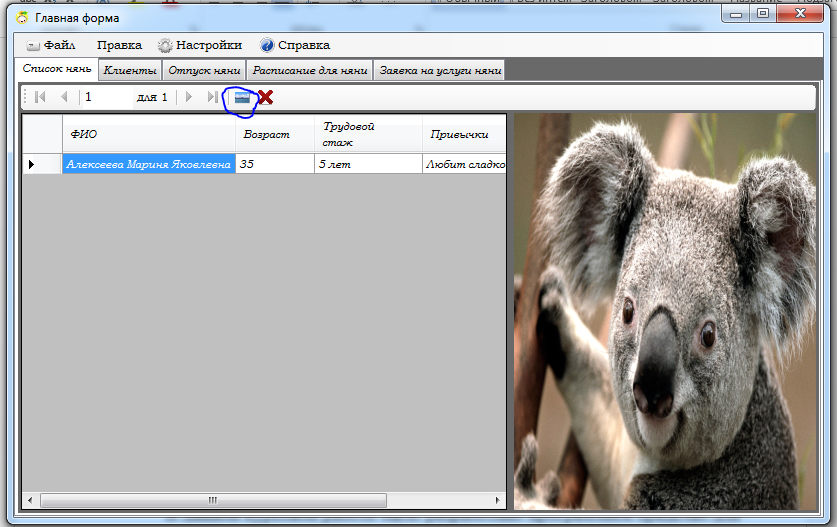


Рисунок 5.6 – добавление фотографии няни

**Заключение**

В данной курсовой работе было разработано программное средство для организации бэбиситерства.

Тестирование всего приложения проводилось методом ручного тестирования путём многократных запусков, разнообразных оправляемых событиях и моделируемых ошибочных ситуаций.

В ходе тестирования были выявлены ошибки в проектировании системы, места в программе, которые замедляют ее. Все данные недочеты были исправлены, что подтверждается в результате многочисленных тестовых запусках приложения.

# 

# 

# Список используемых источников

1. Начальный курс С и С++. Учебник. /Б. И. Березин. Москва:"ДИАЛОГ-МИФИ",1999г.
2. Язык программирования С++. Учебник. /. Страуструп. Киев:"ДиаСофт", 1993 г.
3. Структуры и алгоритмы обработки данных: Учебник. / Матьяш В.А., Путилов В.А., Фильчаков В.В. , Щёкин С.В. - Апатиты, КФ Петр ГУ, 2000.
4. С++ /Дэвис Стефан Р.,4-е издание : Пер. с англ.:- М.: Издательский дом «Вильямс»,2003.
5. Основы программирования: Учеб. Для сред. проф. образования /И.Г.Семакин, А.П.Шестаков. – М., 2006.
6. С++ экспресс курс: Учебник. /Лаптев В.В. – СПб.: БХВ- Петербург 2004.
7. С++ учебный курс: Учебник. /Франка П. – СПб.: Питер 2005.
8. МОДЕЛИ И CТРУКТУРЫ ДАННЫХ:/ Учебное пособие/
9. Д.Далека, А.С. Деревянко, О.Г.Кравец, Л.Е. Тимановская -Харьков: ХГПУ, 2000.