Министерство образования и науки

Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего профессионального образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

Пояснительная записка к приложению ContactApp

По дисциплине

«Новые технологии в программировании»

Выполнил: Студент гр. 581–2

Шаламов А. О.

«\_\_» \_ \_ 2020 г.

Принял: доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_\_Горяинов А. Е.

« » 2020 г.

Томск 2020

**Содержание**

[1 Назначение приложения 3](#_Toc61448485)

[2. Группы пользователей и их функциональные возможности в приложении 3](#_Toc61448486)

[3. Стек технологий разработки. Системные требования 4](#_Toc61448487)

[4 Пользовательский интерфейс 4](#_Toc61448488)

[5 Диаграммы пакетов приложения 8](#_Toc61448489)

[6 Диаграммы классов приложения 8](#_Toc61448490)

[7 Описание тестирования приложения 9](#_Toc61448491)

[8 Описание сборки установщика 11](#_Toc61448492)

[9 Описание модели ветвления в репозитории проекта 13](#_Toc61448493)

# 1 Назначение приложения

Пользовательское приложение ContactApp, предназначено для ведения и хранения контактов.

# 2 Группы пользователей и их функциональные возможности в приложении

Приложение рассчитано на людей, кто знаком с введение контактов в мобильном телефоне.

1) Обеспечивать стабильную работу приложения при порядке 200 контактов.

2) Обеспечивать поиск, навигацию по созданным контактам по фамилии и имени.

3) Предоставить инструменты для просмотра и редактирования контактов.

4) Сохранять и восстанавливать контакты между сессиями приложения.

5) Выполнять промежуточные сохранения контактов на машине пользователя на случай аварийного завершения программы, отключения компьютера и т.д. – для защиты от потери данных.

# 3 Стек технологий разработки. Системные требования

Для проекта взят язык C#. Для тестирования приложения была выбрана библиотека NUnit. Системные требования к приложению ограничены требованиями к .NET Framework 4.6.1.

# 4 Пользовательский интерфейс

После запуска приложения перед пользователем появляется главное окно (рис. 4.1). Двухколоночная верстка главного окна содержит список всех контактов в левой панели и отображает текущий выбранный контакт в правой панели. В списке контактов показаны фамилии контактов, в один момент времени может быть выбран только один контакт (далее – текущий контакт).

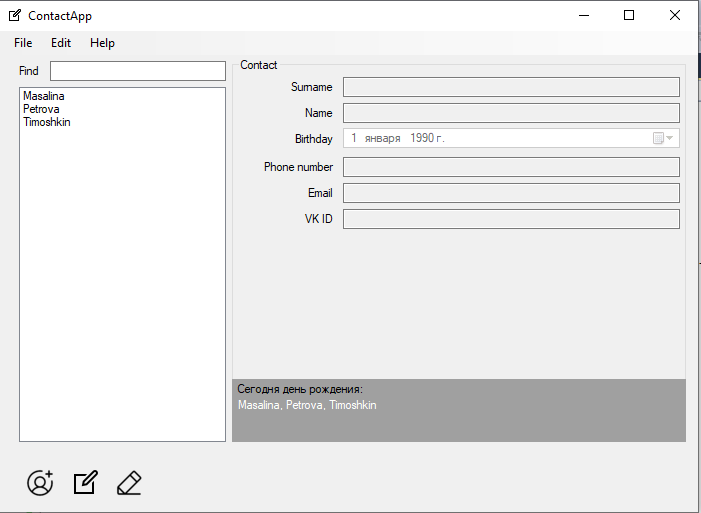


Рис.4.1 – Главное окно приложения ContactsApp

На панели со списком контактов внизу располагаются три кнопки в виде пиктограмм: Add Contact («Создать новый контакт»), Edit Contact («Редактировать текущий контакт»), Remove Contact («Удалить текущий контакт»).

При выборе контакта в списке, выбранный контакт отображается в правой панели. Главное окно не позволяет редактировать содержимое контакта – только просмотр. Также в правой панели под текущим контактом отображается информационная панель с сегодняшними именинниками

При нажатии на кнопку Add Contact и Edit Contact появляется окно создания/редактирования контакта в диалоговом режиме (рис. 4.2). Для нового контакта окно изначально не заполнено (установлена лишь дата рождения по умолчанию). Для редактирования уже существующего контакта все поля должны быть предзаполнены данными текущего контакта.

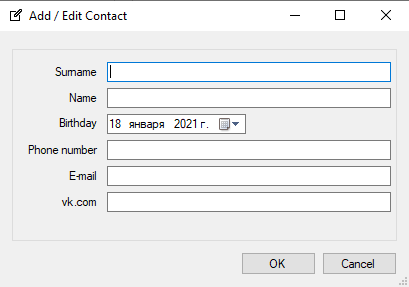


Рис.4.2 – Окно добавления/редактирования контакта в приложении ContactsApp

При нажатии на кнопку OK окно создания контакта закрывается, в список контактов главного окна добавляется новый контакт. При редактировании текущей контакта, нажатие на кнопку OK должно обновить фамилию контакта в списке контактов (если фамилия текущего контакта была изменена/исправлена), и обновить отображаемый контакт в правой панели приложения. При нажатии кнопки Cancel создание/редактирование контакта отменяется (новый контакт не добавляется, исходный контакт остается без изменений).

В случае ввода пользователем некорректных данных (нарушение допустимой длины фамилии, имени, указание невозможной даты рождения или неправильного номера телефона), данная ситуация должна быть обработана соответствующим образом.

При нажатии на кнопку Remove Contact главного окна текущий контакт удаляется. Перед удалением должно появиться окно с запросом на разрешение удаления записи: «Do you really want to remove this contact:<Фамилия текущего контакта>». При нажатии на кнопку OK происходит удаление, при нажатии на кнопку Cancel удаление отменяется.

Меню главного окна содержит следующие пункты:

* File:

• Exit (Выйти из приложения – Alt+F4)

* Edit:

• Add Contact (Создать новый контакт)

• Edit Contact (Редактировать текущий контакт)

• Remove Contact (Удалить текущий контакт)

* Help

• About (Вызвать окно «О программе» - F1) (см. рис. 4.3)

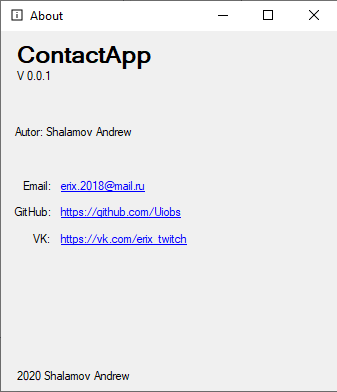


Рис.4.3 – Окно «About» приложения ContactsApp

Таким образом, в программе должно быть реализовано три окна:

- Главное окно.

- Окно создания/редактирования контакта.

- Окно «About».

Верстка главного окна и окна создания/редактирования контакта должна быть адаптивной. Окно «About» имеет фиксированный размер.

Загрузка контактов осуществляется при запуске программы до вывода главного окна пользователю, сохранение контактов в файл должно выполняться в случаях: а) создания нового контакта; б) удаления контакта; в) закрытии приложения.

# 5 Диаграммы пакетов приложения

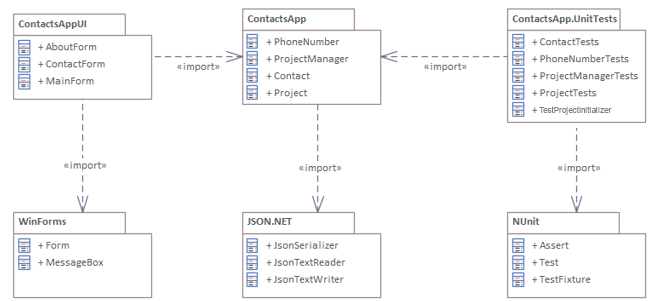


Рис.5.1 – Диаграмма пакетов приложения

# 6 Диаграммы классов приложения

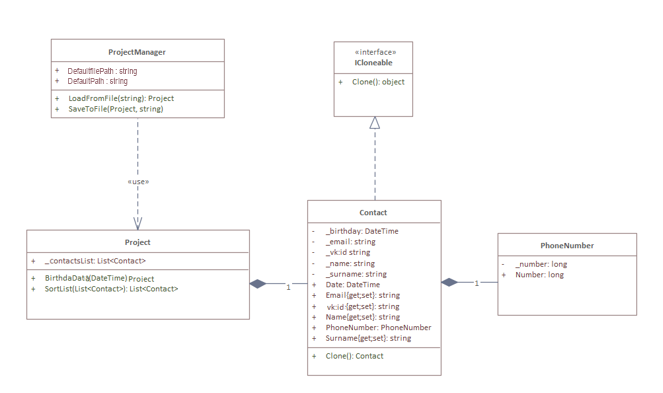


Рис.6.1 – Диаграмма классов приложения

# 7 Описание тестирования приложения

В данном приложении проводились только юнит-тесты. Были написаны тесты, покрывающие код бизнес-логики на 84%. Тесты и их покрытия представлены на рис. 7.1 и 7.2

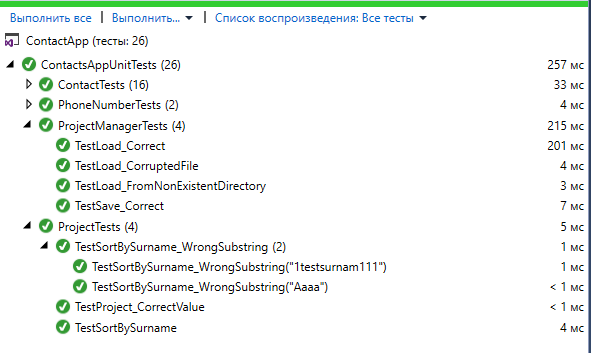


Рис.7.1 – Результаты работы юнит-тестов

Так же перед приемочном тестированием нужно пройти несколько пунктов.

1) Выполнить процедуру непосредственно из-под среды разработки – разработчик убеждается, что последние изменения в программе не внесли ошибок в базовую функциональность и приложение функционирует. Собирается защищенный установщик программы (возможно, с обфускацией кода и внедрением системы лицензирования).

2) Защищенный установщик устанавливается на «чистую» виртуальную машину тестировщиком и полностью проверяется процедура приёмочного тестирования всей системы целиком. Если никаких ошибок в установленном из защищенного установщика приложении не обнаружено, можно приглашать представителя заказчика.

3) Защищенный установщик устанавливается на «чистую» виртуальную машину перед представителем заказчика и полностью выполняется процедура приёмочного тестирования. В завершении, при желании заказчика, ему дают возможность самостоятельно поработать в приложении.

# 

# 8 Описание сборки установщика

Для сборки установщика приложения было использована программа Inno Setup. С его помощью создается установочный сценарий (приложение А), который создает установщик.

В установочном скрипте указывается название программы, номер версии, название компании, указывается в какой папке будет создан установщик, и указаны файлы для работы приложения, такие как .exe и .dll остальные созданные при разработке файлы не нужны (.xml, .config, .pdb, и другие возможные файлы).

В проекте ContactsAppUI

Pre-build:

rmdir /s /q $(SolutionDir)InstallScript\Realese\

Post-build:

md $(SolutionDir)InstallScript\Realese\

xcopy $(ProjectDir)$(OutDir)\*.dll $(SolutionDir)InstallScript\Realese\

xcopy $(ProjectDir)$(OutDir)\*.exe $(SolutionDir)InstallScript\Realese\

В проекте Installer

; Script generated by the Inno Setup Script Wizard.

; SEE THE DOCUMENTATION FOR DETAILS ON CREATING INNO SETUP SCRIPT FILES!

#define MyAppName "Contacts"

#define MyAppVersion "1.0.0.0"

#define MyAppPublisher "Andrew Shalamov"

#define MyAppURL "https://github.com/Uiobs"

#define MyAppExeName "ContactAppUI.exe"

[Setup]

; NOTE: The value of AppId uniquely identifies this application. Do not use the same AppId value in installers for other applications.

; (To generate a new GUID, click Tools | Generate GUID inside the IDE.)

AppId={{02105A29-5B37-4735-8D27-14449FE83237}

AppName={#MyAppName}

AppVersion={#MyAppVersion}

;AppVerName={#MyAppName} {#MyAppVersion}

AppPublisher={#MyAppPublisher}

AppPublisherURL={#MyAppURL}

AppSupportURL={#MyAppURL}

AppUpdatesURL={#MyAppURL}

DefaultDirName={autopf}\{#MyAppName}

DisableProgramGroupPage=yes

LicenseFile={#SourcePath}Realese\LICENSE

; The [Icons] "quicklaunchicon" entry uses {userappdata} but its [Tasks] entry has a proper IsAdminInstallMode Check.

UsedUserAreasWarning=no

; Uncomment the following line to run in non administrative install mode (install for current user only.)

;PrivilegesRequired=lowest

PrivilegesRequiredOverridesAllowed=commandline

OutputBaseFilename=Setup

Compression=lzma

SolidCompression=yes

WizardStyle=modern

[Languages]

Name: "russian"; MessagesFile: "compiler:Languages\Russian.isl"

[Tasks]

Name: "desktopicon"; Description: "{cm:CreateDesktopIcon}"; GroupDescription: "{cm:AdditionalIcons}"; Flags: unchecked

[Files]

Source: "Realese\ContactAppUI.exe"; DestDir: "{app}"; Flags: ignoreversion

Source: "Realese\ContactApp.dll"; DestDir: "{app}"; Flags: ignoreversion

Source: "Realese\Newtonsoft.Json.dll"; DestDir: "{app}"; Flags: ignoreversion

; NOTE: Don't use "Flags: ignoreversion" on any shared system files

[Icons]

Name: "{autoprograms}\{#MyAppName}"; Filename: "{app}\{#MyAppExeName}";

Name: "{autodesktop}\{#MyAppName}"; Filename: "{app}\{#MyAppExeName}"; Tasks: desktopicon

[Run]

Filename: "{app}\{#MyAppExeName}"; Description: "{cm:LaunchProgram,{#StringChange(MyAppName, '&', '&&')}}"; Flags: nowait postinstall skipifsilent

# 9 Описание модели ветвления в репозитории проекта

При разработке приложения использовалась система версионного контроля Git. Ссылка на репозиторий: https://github.com/Uiobs/-

Разработка приложения проводилось в среде Visual Studio.

В репозитории было создано 2 ветки:

Develop – ветка, в которой проводились все разработки.

Master – ветка, содержащая готовый и проверенный код.

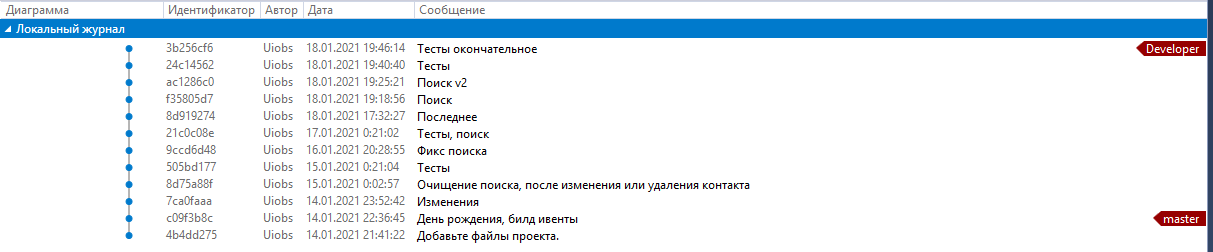


Рис.9.1 – История комментариев ветки develop.