1. **Введение**
   1. Назначение

Приложение предназначено для ежегодного анализа учащихся и их последующего хранения на компьютере.

* 1. Обзор системы

Приложение создано для пользователей ОС Windows.

* 1. Значения и акронимы

Пользователь – человек, который использует продукт.

Сериализация - процесс перевода какой-либо структуры данных в последовательность битов.

Десериализации - восстановление первоначального состояния структуры данных из битовой последовательности.

1. **Предполагаемый дизайн**
   1. Предположения

При запуске приложения появляется главное меню, через которое пользователь может выбрать нужный ему класс, удалить и создать класс. При выборе класса пользователь увидит таблицу учащихся и их оценки по предметам. В этом окне будет доступно добавление, удаление, редактирование, поиск, сортировку учеников. Можно будет посмотреть статистику класса и проверить ученика на склонность к факультативу.

* 1. Ограничения

О таковых информации нет.

* 1. Системная среда

Приложение работает на ОС Windows, для запуска готового приложения необходим только exe-файл.

* 1. Риски

Таковых не выявлено.

1. **Архитектура**
   1. Для написания данного проекта используется парадигма функционального программирования. А именно используются высокоуровневые абстракции, которые скрывают большое количество подробностей таких рутинных операций. За счет этого код получается короче, и, как следствие, гарантирует меньшее количество ошибок, которые могут быть допущены.
   2. Стратегии и решения

На архитектуру повлияли следующие решения

и стратегии:

1. В коде используются lambda функции
2. Интерфейс взаимодействуюет с пользователем через функции высшего порядка
3. **Дизайн высокого уровня**
   1. Описание

Основные компоненты – menu, class\_win , find\_student\_and\_analysis , stats\_win, search\_win, info\_win.

Компоненты взаимодействуют между собой посредством функций высшего порядка.

**Окно меню:**

В самом компоненте меню находятся кнопки перехода к классу, а так же выпадающие кнопки. При нажании на выпадающую “Меню” на экране появятся кнопки удаления, добавления, отмены удаления, выхода. При нажатии на удаление, напротив классов появятся кнопки “удалить”. Для отмены необходимо выбрать пункт “отмена удаления”

**Окно класса:**

При переходе на окно класса, пользователь видит таблицу с учениками и их оценками. Над каждым столбцом отметок имеется кнопка названия предмета, если на неё нажать столбец будет отсортирован. Над учениками имеется такая же кнопка. Сверху находится выпадающее меню, где с помощью которого можно удалять, редактировать, искать учеников. Также в этом меню можно выбрать пункт анализа класса.

**Окно анализа:**

При выборе анализа оценки в выпадающем меню. Появится окошко с выбором предмета для анализа. Выбрав предмет, на экране появится возрастающий график оценок по выбранному предмету. Далее этот график сохранится в папке.

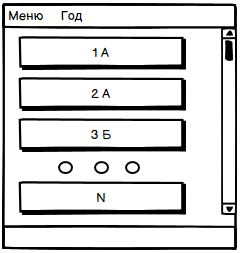
1. **Структура хранения данных**

5.1 Описание

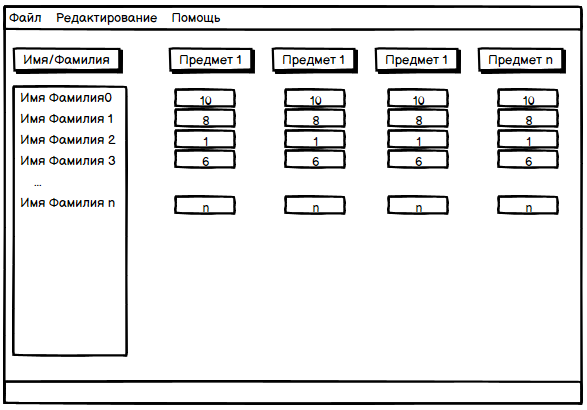
Для хранения данных был выбран модуль pickle. Модуль pickle реализует мощный алгоритм сериализации и десериализации объектов Python. "Pickling" - процесс преобразования объекта Python в [поток байтов](https://pythonworld.ru/tipy-dannyx-v-python/bajty-bytes-i-bytearray.html), а "unpickling" - обратная операция, в результате которой поток байтов преобразуется обратно в Python-объект. Так как поток байтов легко можно [записать в файл](https://pythonworld.ru/tipy-dannyx-v-python/fajly-rabota-s-fajlami.html), модуль pickle широко применяется для сохранения и загрузки сложных объектов в Python.

1. **Дизайн низкого уровня**
   1. Страницы приложения

6.2.1 Страница меню



6.2.2 Страница класса



6.2.3 Окно статистики

