ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ» МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ

**ЗАДАНИЕ  
для самостоятельной работы по индивидуальному плану по курсу  
«Проектный семинар “Python в науке о данных”»**

Студент группы БИВ234:

* Шарифулин Максим Фаритович, mfsharifulin@edu.hse.ru, +79172832656

Тема работы:

Приложение для отслеживания КБЖУ питания и рекомендательная система

Руководитель: Поляков Константин Львович

**Москва 2024**

Участники договора

Настоящий документ является техническим заданием (далее ТЗ), которое формулируется преподавателем (далее Заказчик) для выполнения студентом (далее Исполнитель).

Права исполнителя

Исполнитель полностью освобождается от посещения занятий и участия в любых контрольных мероприятиях. Оценка за курс исполнителя совпадает с оценкой за проект.

Полноценная работоспособная реализация данного ТЗ высокого качества гарантирует получение оценки девять или десять баллов (отлично). Нарушение сроков и положений ТЗ, а также наличие ошибок ведет к адекватному снижению оценки.

Решаемая задача

Исполнитель создает приложение на платформе Windows 10 с графическим интерфейсом, именуемое клиентом, которое служит для отслеживания питания и показ БЖУ за неделю. На основе данной информации приложение будет генерировать рецепты, подходящие для поддержания БЖУ

Вычислительная платформа и основные библиотеки

Для реализации задачи используется язык программирования Python 3 в составе обязательного для курса дистрибутива Anaconda.

Также будут использованы следующие библиотеки:

* TKinter, Custom TKinter
* Matplotlib для формирования графиков
* DateTime, Time
* JSON

Функционал приложения

Приложение обеспечивает:

1. Подсчет БЖУ за последнюю неделю
2. Рекомендации по питанию

Архитектура приложения и требования к коду

Приложение должно запускаться из командной строки запуском на счет основного модуля или командного файла. Установка приложения, в частности создание при необходимости виртуальной среды, должна быть полностью автоматизирована, т.е. осуществляться запуском на счет модуля Python или командного файла. Установка приложения и его эксплуатация не должны требовать выполнения системных операций, связанных с изменениями операционной системы, актуального дистрибутива Anaconda и прочих элементов вычислительной среды пользователя. В частности, развертывания ПО Docker и аналогичных инструментов.

Для работы приложения должна быть создана виртуальная среда исключительно средствами менеджера пакетов conda из актуального дистрибутива Anaconda. Таким образом, приложение должно работать в собственной виртуальной среде на любом компьютере, соответствующем требованиям данного документа, без необходимости добавления библиотек к актуальному дистрибутиву Anaconda. Приложение является локальным приложением с графическим интерфейсом, которое работает на изолированном компьютере. Это означает, что для работы приложения не нужны вычислительные ресурсы Интернета и не используется клиент-серверная архитектура (за исключением Web-интерфейса). В тоже время, приложение может анализировать данные из Интернета. Также приложение должно исполняться на компьютере средней мощности с 8Gb оперативной памяти и 2Gb видеопамяти не очень мощной видеокарты. Приложение реализовано в виде основного модуля и библиотеки, оформленной в соответствии с требованиями Python и графического интерфейса, который использует функции данной библиотеки. Библиотека образована модулями, содержащими тематически близкие функции/классы. Каждый модуль содержит не более 300 строк. Код разработан в соответствии с требованиями «Python Enhancement Proposals (PEP) 8 -- Style Guide for Python Code» и «PEP 257 -- Docstring Conventions» (см. https://www.python.org/dev/peps/).

Описание интерфейса

Приложение имеет графический интерфейс, адаптированный для реализации указанного выше функционала. В частности,

1. Инфографики БЖУ за неделю и норма
2. Генерация нескольких блюд для поддержания нормы БЖУ
3. Регистрирование новой пищи для учета БЖУ

Сопроводительная документация

Исполнитель создает два документа: «Руководство пользователя», содержащее подробное описание решаемой задачи, инсталляции, запуска и интерфейса приложения, и «Руководство разработчика», содержащее подробное описание архитектуры приложения — состав библиотек, функций и классов, связь между ними, структура каталогов и описание параметров, управляющих работой приложения. Документация формируется в виде двух файлов формата MS Word 2016 или Adobe Acrobat.

Сроки предоставления результатов

Здесь приводится план-график выполнения самостоятельной работы. Самостоятельная работа разделяется на части. Каждая часть представляется к определенному сроку и оценивается по принципу «зачет/не зачет». Приступать к разработке следующей части без получения зачета по предыдущей не разрешается.

План-график:

|  |  |
| --- | --- |
| Дата | Этап |
| 28.02.2024 | Анализ норм БЖУ, составление возможных планов питания |
| 28.03.2024 | Первичная разработка интерфейса, проектирование архитектуры приложения |
| 10.04.2024 | Интеграция системы с СУБД, сохранение и обработка данных |
| 09.05.2024 | Разработка итогового продукта, написание интерфейса и системных модулей |
| 09.06.2024 | Отладка, тестирование. Написание «Руководства пользователя», «Руководства разработчика» |

Критерии оценивания

• Работоспособность приложения. При запуске и использовании функционала, указанного в «Руководстве пользователя» не должно возникать ошибок и не должно выводятся в консоль сообщений, связанных с некорректной работой приложения. Возможности языка Python и специально подобранные требования к разрабатываемому продукту позволяют с уверенностью утверждать, что следование указаниям задания для самостоятельной работы гарантирует работоспособность создаваемого при выполнении задания приложения на любом компьютере, на котором установлена общая для всех студентов версия дистрибутива Python — Anaconda, при отсутствии ошибок разработчиков данного приложения. Таким образом, если разработчик корректно создал код приложения оно должно работать, как на его компьютере, так и на компьютере другого пользователя, в частности, преподавателя, при условии выполнения указанных выше требований. Отсутствие возможности приступить к эксплуатации приложение может свидетельствовать только, либо о наличии ошибок в коде, либо о нарушении требований задания, в частности, использовании специфических особенностей компьютера разработчика. Отсутствие возможности приступить к работе с приложением (критическая ошибка1 при запуске) или прекращение работы приложения по причине возникновения критической ошибки в ходе его эксплуатации автоматически приводит к оценке **ноль баллов по десятибалльной системе**. Наличие не критических ошибок приводит к адекватному снижению балла за приложение.

* Структура приложения. Полнота и качество реализации функционала и интерфейса, описанных в настоящем документе. Приложение должно соответствовать всем требованиям, указанным в настоящем документе. Отсутствие тех или иных требуемых элементов приводит к адекватному снижению балла за приложение. Полное несоответствие требованиям приводит к оценке **ноль баллов по десятибалльной системе**.
* Интерфейс. Качество реализации и удобство работы с интерфейсом всех частей приложения — минимально возможное наличие ручного ввода данных, интуитивная понятность и удобство размещения элементов графического интерфейса (виджетов), возможность конфигурирования интерфейса (настройки цветов и шрифтов, а также управления окнами с помощью мыши). Минимально допустимой реализацией настройки интерфейса является использование конфигурационных файлов. Приветствуется наличие графического интерфейса для выполнения настроек. Отсутствие возможности настраивать интерфейс является нарушением требований и приводит к оценке **ноль баллов по десятибалльной системе**.
* Реализация кода. Код должен быть разработан в соответствии с требованиями «Python Enhancement Proposals (PEP) 8 -- Style Guide for

1 Критической ошибкой называется ошибка работы кода приложения, которая делает невозможным его дальнейшую эксплуатацию.

Python Code» и «PEP 257 -- Docstring Conventions» (см.

https://www.python.org/dev/peps/). Оценка качества кода в IDE Spyder должна быть не ниже 7 баллов. Более низкая оценка приводит к оценке **ноль баллов по десятибалльной системе**. Более высокий балл является аргументом для повышения оценки. Код должен содержать много комментариев, поясняющих назначение всех его частей. Количество и качество комментариев влияет на оценку.

• Документация к приложению. Оценивается качество реализации документации — «Руководства пользователя» и «Руководства разработчика». Документы должны иметь пронумерованные страницы и разбиты на разделы. В документах должно присутствовать оглавление, созданное средствами MS Word. При реализации в формате Adobe Acrobat, оглавление должно обеспечивать навигацию по документу. Технические требования: шрифт Calibri Light 12, межстрочный интервал 1.15, все таблицы и рисунки (скриншоты) пронумерованы и имеют название, каждый документ имеет титульный лист с указанием названия учебного заведения, учебной программы, учебной группы, номера и состава бригады (приведен телефон и адрес Директора).

Плагиат. За исключением кода используемых библиотек, служебного кода необходимого для их работы, а также кода, созданного библиотеками в процессе их работы, совпадение пятидесяти и более процентов строк кода с кодом, опубликованным в открытых источниках или кодом ранее выполненных индивидуальных проектов по данному курсу, считается плагиатом, и Исполнитель получает оценку **ноль баллов по десятибалльной системе.**