Ответы на вопросы:

0) Неформализованное описание задачи:

Необходимо реализовать программу, которая визуализирует успеваемость студента, группы, потока, факультета (баллы должны высчитываться с учетом степени важности предметов).

1) Название: Universal Rating Visualizer (универсальный визуализатор рейтинга).

2) Цель: дать подходящее визуальное отображение рейтинга студента, группы, потока или факультета. Под подходящим визуальным отображением рейтинга понимается сопоставление среднему баллу студента, группы, потока или факультета некоему изображению, отражающему успеваемость обучающейся единицы в интересующий нас промежуток времени (за первую контрольную неделю, за вторую контрольную неделю или за весь семестр). Рейтинг группы представляет собой среднее арифметическое среди всех студентов группы. Аналогичным образом рейтинг потока - есть среднее арифметическое рейтингов входящих в него групп с одного курса, а рейтинг факультета - среднее арифметическое всех потоков, входящих в факультет.

Назначение: визуализация успеваемости студента, группы или факультета на основе данных рейтинга.

3) Область применения ПО: интернет-ресурсы, связанные с отображением рейтинга студентов ВолгГТУ (системы рейтинг-контроля студентов ВолгГТУ).

4) В ходе беседы с заказчиком нами было найдено два реально существующих аналога.

Первый «Когнитивный визуализатора рейтинга». Данная программа представляет рейтинг студента в виде колоды карт. Градация идет как по виду карты, так и по масти.

Минус данной системы состоит в том, что она способна визуализировать рейтинг только за егэ, контрольные недели и экзамен. Нет таких промежуточных срезов как контрольные и курсовые работы.

Также система визуализации в виде колоды карт несовсем понятна. Масти карт вряд ли могут адекватно отразить важность того или иного среза.

Неудобная система отображения рейтинга нескольких студентов, непонятно, к кому какая карта, более того, от карт рябит в глазах, от чего становится очень затруднительно разобраться, к какому студенту какая карта относится.

Второй «Когнитивный рейтинг студента в виде "лица Чернова"». Программа представляет собой апплет на языке Java и предназначена для

отображения 15 параметров в виде «лица Чернова».

Достоинством данной программы можно считать то, что в ней каждый предмет отвечает за отдельный элемент «лица Чернова», что дает нам множество разнообразных состояний выражения лица.

Так же значения важности (максимальные/минимальные) пользователь может произвольно задать для каждого предмета самостоятельно.

В качестве недостатков стоит отметить то, что не всегда понятно, что выражает «лицо Чернова», т.к. отдельные элементы лица могут не сочетаться с остальными по настроению, что портит общую картину и выглядит неестественно.

Также недостатком может являтся возможность отображения лишь 15 параметров, что не всегда подходит для построения рейтинга.

Помимо этого, нет системы отображения рейтинга для нескольких студентов, групп или факультетов.

5) Протокол извлечения знаний:

1. В какой сайт будет интергрироваться разработанное ПО? На чем он основан(платформа, язык, CMS, дополнительное ПО)?
2. Размер шкалы(количество цветов/размеров рыб/чинов ангелов и демонов/ видов энтов)?
3. Источники информации, откуда можно извлечь данные об успеваемости студентов? В каком виде? Какие для этого нужны права?
4. Подойдут ли цвета для отображения, что использовать по вашему мнению?
5. Визуализация каких промежуточных результатов студентов помимо контрольных недель, экзаменов и итоговых результатов Вас интересуют? (отдельные виды контрольных работ, лабораторные работы и тд)
6. Нужно ли вносить визуализацию рейтинга у преподавателей? Если да, то по каким критериям высчитывается рейтинг преподавателей?
7. Как в университете определяется профильность предмета? Имеет ли величина ЗЕТ к этому прямое отношение?
8. Что делать с курсовыми проектами, которые выставляются отдельно? Нужна ли визуализация и для них отдельно?
9. Если мы будем брать данные с БД vstu.ru(или другого сервера), нужен доступ к БД и ее структура, также какие данные отсутствуют в этой БД, например, есть ли там сведения о предметах? Или нужно изменить подход и создавать собственную БД?

6) Роли пользователей:

1. Анонимный пользователь - незарегистрированный пользователь, который может только просматривать рейтинг.

2. Модератор - человек, вносящий данные об успеваемости студентов в систему (например, староста группы).

3. Администратор - человек, проверяющий работу модератора. Он может принять или отклонить введенные модератором данные. Также способен сам вносить нужные данные в систему (например, преподаватель).

7) Описание объекта проектирования (то, что делает программа):

программа производит визуализацию данных системы рейтинг-контроля для выбранного студента, группы или факультета. Отображаемые данные состоят из рейтинга студента/группы/факультета по одному или нескольким предметам, или успеваемости студента за всё время обучения.

8) Описание функциональных требований к программе:

1. Программа должна выводить данные о рейтинге студента, группы, потока, факультета в понятной визуальной форме.

2. Программа должна уметь классифицировать имеющиеся предметы по трем группам (профильные, дополнительные, смежные)

3. Программа должна уметь высчитывать средний рейтинг студента, группы, потока и факультета, учитывая степень профильности предмета.

4. Программа должна обеспечивать возможность сравнивать между собой рейтинги нескольких равнозначных обучаемых единиц.

9) Описание нефункциональных требований:

1. Программа должна корректно реагировать на ежегодное изменение количества студентов и предметов.

10) Входные и выходные данные:

Входные данные: успеваемость (рейтинг) студентов, скорее всего, взятые из базы данных.

Выходные данные: визуальное представление рейтинга студента, группы, факультета поодиночке или в сравнении.

11) Ограничение границ проекта:

Визуализатор должен корректно работать с текущим количеством студентов, обучающихся в ВолгГТУ, а также с полным набором предметов, имеющихся в учебном плане.

Визуализатор должен работать с тремя группами предметов (профильные, дополнительные, смежные) и не более того.

Также программа должна уметь производить сравнение рейтингов следующих видов: “студент - студент”, “группа - группа”, “поток - поток” и “факультет - факультет” и никаких других. Сравнение рейтингов должно происходить между равнозначными обучающимися единицами.

12) Роли в команде:

Клевцов В. -- руководитель/тех. райтер/кодер

Никитский Н. кодер/архитектор

Пестун М. -- ???

Поцелуйко А. -- кодер/тестер/верстальщик/тех. райтер