**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| УТВЕРЖДАЮ  Руководитель (должность, наименование предприятия) – | | | | УТВЕРЖДАЮ  Руководитель (должность, наименование предприятия) – | | | |
| Личная подпись |  | Расшифровка подписи |  | Личная подпись |  | Расшифровка подписи |  |
| Печать |  | | | Печать |  | | |
| Дата |  | | | Дата |  | | |

**Информационная система «Приложение по учету успеваемости студентов»**

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

На 23 листах

**Действует с « 17 » ноября 2021 г.**

|  |  |  |  |
| --- | --- | --- | --- |
| СОГЛАСОВАНО | | | |
| Руководитель (должность, наименование предприятия) – | | | |
| Личная подпись |  | Расшифровка |  |
| Печать | |  |  |
| Дата | |  |  |

**Москва 2021**

**Содержание**

[1 ОБЩИЕ ПОЛОЖЕНИЯ 4](#_Toc88586334)

[1.1 Наименование системы 4](#_Toc88586335)

[1.2 Наименование организации – Заказчика и Разработчика 4](#_Toc88586336)

[1.3 Плановые сроки начала и окончания работы 4](#_Toc88586337)

[2 ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ 5](#_Toc88586338)

[3 ХАРАКТЕРИСТИКА ОБЪЕКТОВ АВТОМАТИЗАЦИИ 6](#_Toc88586339)

[3.1 Описание предметной области 6](#_Toc88586340)

[3.2 Описание технологии обработки информации 7](#_Toc88586341)

[4 ТРЕБОВАНИЯ К СИСТЕМЕ 10](#_Toc88586342)

[4.1. Требования к языкам программирования 10](#_Toc88586343)

[4.2 Требования к безопасности 10](#_Toc88586344)

[5 СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ СИСТЕМЫ 11](#_Toc88586345)

[5.1 Добавление студента 11](#_Toc88586346)

[5.2 Редактирование данных о студенте 11](#_Toc88586347)

[5.3 Добавление задания 11](#_Toc88586348)

[5.4 Редактирование данных о задании 12](#_Toc88586349)

[5.5 Добавление группы 12](#_Toc88586350)

[5.6 Редактирование данных о группе 12](#_Toc88586351)

[5.7 Добавление дисциплины 13](#_Toc88586352)

[5.8 Редактирование данных о дисциплине 13](#_Toc88586353)

[5.9 Добавление занятия 13](#_Toc88586354)

[5.10 Выставление оценки 14](#_Toc88586355)

[5.11 Анализ студентов 14](#_Toc88586356)

[5.12 Формализация критериев поиска студентов с задолженностями 16](#_Toc88586357)

[5.13 Формализация критериев поиска наиболее успевающих студентов 16](#_Toc88586358)

[5.14 Формализация критериев сортировки студентов по количеству задолженностей, количеству предпринятых попыток сдачи и по преуспевающим программу студентам 17](#_Toc88586359)

[5.15 Формализация критериев сортировки задолженностей конкретного студента 17](#_Toc88586360)

[6 ПОРЯДОК КОНТРОЛЯ И ПРИЁМКИ СИСТЕМЫ 19](#_Toc88586361)

[7 ТРЕБОВАНИЯ К СОСТАВУ И СОДЕРЖАНИЮ РАБОТ ПО ПОДГОТОВКЕ ОБЪЕКТА АВТОМАТИЗАЦИИ К ВВОДУ СИСТЕМЫ В ДЕЙСТВИЕ 20](#_Toc88586362)

[8 ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ 21](#_Toc88586363)

[8.1. Общие требования к документированию 21](#_Toc88586364)

[8.2. Перечень подлежащих разработке документов 21](#_Toc88586365)

[9 ПОРЯДОК ВНЕСЕНИЯ ИЗМЕНЕНИЙ 22](#_Toc88586366)

[ПРИЛОЖЕНИЕ 3 23](#_Toc88586367)

# 1 ОБЩИЕ ПОЛОЖЕНИЯ

## 1.1 Наименование системы

**1.1.1 Полное наименование системы**

Информационная система «Приложение по учету успеваемости студентов».

## 1.2 Наименование организации – Заказчика и Разработчика

**1.2.1 Заказчик**

Заказчик: Красников Александр Сергеевич

Адрес фактический: 107023, г. Москва, ул. Б.Семёновская, д. 37.

Телефон / Факс: +7 (800) 555-35-35.

**1.2.2 Разработчик**

Разработчик: ИП «Герасименко Щеголькова Поляков», Адрес фактический: 111111, г. Москва, ул. Электрозаводская, д. 1. Телефон / Факс: +7 (999) 999-99-99.

## 1.3 Плановые сроки начала и окончания работы

Плановый срок начала работ по созданию информационной системы «Приложение по учету успеваемости студентов» ­­­­­­­­­— 17 ноября 2021 года.

Плановый срок окончания работ по созданию информационной системы «Приложение по учету успеваемости студентов» — 31 декабря 2021 года.

**2 ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ**

Объектом изучения данной курсовой работы является деятельность преподавателя в университете.

Предметом изучения является анализ и контроль функциональной деятельности учета успеваемости студентов в университете.

Цель работы: автоматизировать учет успеваемости студентов.

Задачи работы:

1. Исследование предметной области;

2. Разработка модели учета и контроля успеваемости студентов;

3. Написание программного продукта в среде выполнения программ CLR на платформе Microsoft.NET Framework.

4. Тестирование системы «Автоматизация учета успеваемости студентов».

## 3 ХАРАКТЕРИСТИКА ОБЪЕКТОВ АВТОМАТИЗАЦИИ

## 3.1 Описание предметной области

Каждому студенту университета присваивается уникальный номер (номер зачетной книжки), по которому можно однозначно идентифицировать его фамилию, имя, отчество, специальность, на которой обучается студент, номер текущего курса обучения, номер его учебной группы, а также получить сведения о его текущей успеваемости.

Для каждой из изучаемых дисциплин может быть запланировано несколько видов занятий (семинары, практические занятия, лабораторные работы и т.д.), на которых оцениваются знания студентов. При учете текущей успеваемости студента указывается изучаемая дисциплина, вид проводимого занятия, тема занятия, порядковый номер темы по плану занятий, срок (дата), к которому работа по данной теме должна быть выполнена. Помимо этого, отмечается номер выполняемого студентом варианта задания по данной теме, количество предпринятых студентом попыток отчитаться по выполненной работе в соответствии с данным вариантом, оценочный балл по результатам выполнения работы.

Одному студенту по конкретной теме выдается только один вариант задания. Студенты одной группы не могут выполнять одинаковые варианты заданий по одной и той же теме для одного вида занятия по одной дисциплине.

На основе анализа текущей успеваемости формируется очередь из студентов, желающих отчитать выполненные работы. Очередь формируется из студентов одной учебной группы, у которых есть задолженности по каким-либо темам, запланированным для определенного вида занятия по определенной дисциплине. На основных занятиях, предусмотренных расписанием, в первую очередь рассматриваются студенты, которые выполнили все задания, кроме текущей (по плану занятий) темы. Затем рассматриваются студенты в порядке возрастания количества задолженностей по предыдущим темам. Студенты, которые опережают план занятий и желают отчитать работу по следующим темам, рассматриваются в последнюю очередь. Из нескольких задолженностей студент выбирает к сдаче тему, которая по плану занятий должна изучаться первой. Среди студентов, имеющих одинаковое количество задолженностей (в том числе ни одной), первыми принимаются те, кто предпринял наименьшее количество попыток отчитать запланированную к сдаче работу.

Преподаватель может выбрать к поощрению нескольких (на свое усмотрение) наиболее успевающих по изучаемой дисциплине студентов. Список предлагаемых к поощрению студентов, формируется из лиц, которые выполнили все задания по данной дисциплине в запланированные сроки. При этом в первую очередь рассматриваются студенты, набравшие наиболее высокий суммарный балл по всем выполненным заданиям.

Таким образом, необходимо автоматизировать процесс учета и контроля успеваемости студентов.

## 3.2 Описание технологии обработки информации

По результатам анализа предметной области была выявлена основная функциональность системы, представленная на диаграмме вариантов использования. Данная диаграмма состоит из одного пользователя (преподаватель), для которого система реализует то или иное поведение, и действий, которые должны быть выполнены системой в ответ на внешние воздействия пользователя.

Пользователь (преподаватель) может выполнять следующие действия: редактировать список групп (добавлять, удалять и изменять), редактировать список дисциплин (добавлять, удалять и изменять), редактировать список студентов (добавлять, удалять и изменять), редактировать список заданий (добавлять, удалять и изменять), редактировать список занятий (добавлять, удалять и изменять), а также анализировать текущую успеваемость студентов и вносить в нее изменения.

При редактировании списка групп система предоставляет возможность просматривать полный список групп. Пользователь может добавить, удалить группу, а также изменить у выбранной группы характеристики (название, специальность, курс, факультет).

При редактировании списка дисциплин система предоставляет возможность просматривать полный список дисциплин всех групп. Пользователь может добавить, удалить дисциплину, а также изменить у выбранной дисциплины характеристики (название, группу).

При редактировании списка студентов система предоставляет возможность просматривать полный список студентов всех групп. Пользователь может добавить, удалить студента, а также изменить его характеристики (фамилия, имя, отчество, группу).

При редактировании списка заданий система предоставляет возможность просматривать полный список заданий по выбранной дисциплине. Пользователь может добавить, удалить задание, а также изменить некоторые его характеристики (название, срок, тип задания, тема задания, максимальный балл).

При редактировании списка занятий система предоставляет возможность просматривать полный список занятий. Пользователь может добавить или удалить, открыв его. Редактирование характеристик занятия невозможно.

Анализ студентов с задолженностями позволяет просмотреть отсортированный список студентов одной группы в порядке возрастания количества задолженностей по предыдущим темам определенной дисциплины. Если количество задолженностей одинаково, то первым в списке будет тот, у кого по планируемой для сдачи задолженности будет меньше попыток сдачи. Данный список составляется для дополнительного занятия.

Общий анализ студентов позволяет посмотреть общий список студентов группы (т.е. в списке присутствуют и те, у кого есть задолженности, и те, у кого их нет). Список составляется для текущего занятия, на котором можно отчитать только одно задание, поэтому список сортируется по возрастанию количества задолженностей; если количество задолженностей совпадает, то первый в списке будет тот, у кого по заданию текущего занятия меньше всего попыток сдачи; в последнюю очередь принимаются те, кто опережает план занятий, т.е. текущая работа у них сдана.

Анализ задолженностей студента проходит на дистанционных занятиях, которые проводятся для одного студента группы. Под задолженностями в данном контексте понимаются все работы, в т.ч. и те, у которых срок сдачи еще не истек. Анализ заключается в сортировке всех несданных заданий по возрастанию срока сдачи.

Анализ наиболее успевающих студентов позволяет посмотреть список студентов, которых преподаватель может поощрить. Список предлагаемых к поощрению студентов, формируется из лиц, которые выполнили все задания по данной дисциплине в запланированные сроки. При этом в первую очередь рассматриваются студенты, набравшие наиболее высокий процент выполнения всех заданий.

Для работы со списками студентов, дисциплин, групп, заданий и занятий необходимо обеспечить доступ к ним. Для этого была реализована возможность работы с файлами, необходимыми для хранения данных.

При открытии программа загружает все необходимые данные из созданных ранее файлов. После этого пользователю будут доступны любые действия, указанные выше.

## 4 ТРЕБОВАНИЯ К СИСТЕМЕ

## 4.1. Требования к языкам программирования

Для разработки приложения использовался язык программирования C#.

## 4.2 Требования к безопасности

Для создания новых (или редактирования старых) записей, необходима авторизация в качестве администратора.

Приложение должно открываться только после аутентификации пользователя.

# 5 СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ СИСТЕМЫ

## 5.1 Добавление студента

При добавлении студента в список пользователю необходимо заполнить поля, содержащие данные о нем.

После заполнения система проверяет полноту введенных данных, то есть проверяется заполнение обязательных полей, таких как фамилия и имя.

В случае корректности ввода данных система производит запрос менеджеру на генерацию уникального идентификационного номера ID для добавляемого студента, затем создает студента и добавляет его в общий список. Таким образом, система должна всегда при добавлении нового студента контролировать присвоение индивидуального номера (ID).

В случае некорректности ввода данных о студенте, система выдает сообщение об ошибке, пользователю будет предложено ввести данные в обязательные поля.

## 5.2 Редактирование данных о студенте

При выполнении данного действия пользователю необходимо сначала выбрать студента, затем изменить поля, содержащие данные о студенте.

После заполнения система проверяет полноту введенных данных, то есть проверяется заполнение обязательных полей, таких как фамилия и имя.

Если данные введены корректно, система изменяет данные о студенте. Иначе система выдает сообщение об ошибке, пользователю будет предложено изменить данные в обязательных полях.

## 5.3 Добавление задания

При выполнении данного действия пользователю необходимо заполнить поля, содержащие данные о задании.

После заполнения система проверяет полноту введенных данных, то есть проверяется заполнение обязательных полей, таких как название, срок, тип задания, тема задания, порядковый номер темы и максимальный балл.

В случае корректности ввода данных система производит запрос менеджеру на генерацию уникального идентификационного номера ID для добавляемого задания, затем создает задание и добавляет его в общий список. Таким образом, система должна всегда при добавлении нового задания контролировать присвоение индивидуального номера (ID).

В случае некорректности ввода данных о задании, система выдает сообщение об ошибке, пользователю будет предложено ввести данные в обязательные поля.

## 5.4 Редактирование данных о задании

При выполнении данного действия пользователю необходимо сначала выбрать задание, затем изменить поля, содержащие данные о нем.

После заполнения система проверяет полноту введенных данных, то есть проверяется заполнение обязательных полей, таких как название, срок, тип задания, тема задания, порядковый номер темы и максимальный балл.

Если данные введены корректно, система изменит данные о задании. Иначе выдаст сообщение об ошибке, пользователю будет предложено изменить данные в обязательных полях.

## 5.5 Добавление группы

При выполнении данного действия пользователю необходимо заполнить поля, содержащие данные о группе.

После заполнения система проверяет полноту введенных данных, то есть проверяется заполнение обязательных полей, таких как название, специальность, курс и факультет.

В случае корректности ввода данных система производит запрос менеджеру на генерацию уникального идентификационного номера ID для добавляемой группы, затем создает группу и добавляет ее в общий список. Таким образом, система должна всегда при добавлении новой группы контролировать присвоение индивидуального номера (ID).

В случае некорректности ввода данных о группе, система выдает сообщение об ошибке, пользователю будет предложено ввести данные в обязательные поля.

## 5.6 Редактирование данных о группе

При выполнении данного действия пользователю необходимо сначала выбрать группу, затем изменить поля, содержащие данные о ней.

После заполнения система проверяет полноту введенных данных, то есть проверяется заполнение обязательных полей, таких как название, специальность, курс и факультет.

Если данные введены корректно, система изменит данные о задании. Иначе выдаст сообщение об ошибке, пользователю будет предложено изменить данные в обязательных полях.

## 5.7 Добавление дисциплины

При выполнении данного действия пользователю необходимо заполнить поля, содержащие данные о дисциплине.

После заполнения система проверяет полноту введенных данных, то есть проверяется заполнение обязательных полей, таких как название и группа.

В случае корректности ввода данных система производит запрос менеджеру на генерацию уникального идентификационного номера ID для добавляемой дисциплины, затем создает дисциплину и добавляет ее в общий список. Таким образом, система должна всегда при добавлении новой дисциплины контролировать присвоение индивидуального номера (ID).

В случае некорректности ввода данных о дисциплине, система выдает сообщение об ошибке, пользователю будет предложено ввести данные в обязательные поля.

## 5.8 Редактирование данных о дисциплине

При выполнении данного действия пользователю необходимо сначала выбрать дисциплину, затем изменить поля, содержащие данные о ней.

После заполнения система проверяет полноту введенных данных, то есть проверяется заполнение обязательных полей, таких как название и группа.

Если данные введены корректно, система изменит данные о дисциплине. Иначе выдаст сообщение об ошибке, пользователю будет предложено изменить данные в обязательных полях.

## 5.9 Добавление занятия

При выполнении данного действия пользователю необходимо сначала выбрать вид занятия, а затем заполнить поля, содержащие данные о нем.

После заполнения система проверяет полноту введенных данных, то есть проверяется заполнение обязательных полей, таких как вид занятия, дата и время занятия, группа, дисциплина, а также, в зависимости от выбранного вида занятия, следующие поля: номер аудитории, задание, студент, ID конференции и пароль конференции.

В случае корректности ввода данных система производит запрос менеджеру на генерацию уникального идентификационного номера ID для добавляемого занятия, затем создает занятие и добавляет его в общий список. Таким образом, система должна всегда при добавлении нового занятия контролировать присвоение индивидуального номера (ID).

В случае некорректности ввода данных о занятии, система выдает сообщение об ошибке, пользователю будет предложено ввести данные в обязательные поля.

## 5.10 Выставление оценки

При добавлении задания система всем студентам, причастным к нему, выставляет нуль-оценку (0 баллов), поэтому выставление оценки, по сути, является ее изменением.

При выполнении данного действия пользователю необходимо выбрать группу, дисциплину, студента и задание, по которому будет выставляться оценка, а затем изменить соответствующее поле.

После заполнения система проверяет полноту введенных данных, то есть проверяется заполнение обязательных полей и некоторые условия (оценка не может быть больше 100 или меньше 5, или больше максимальной оценки за задание).

Если данные введены корректно, система изменит данные об оценке. Иначе выдаст сообщение об ошибке, пользователю будет предложено изменить данные.

## 5.11 Анализ студентов

Анализ студентов проводится с целью формирования ограниченного и упорядоченного по заданным критериям списка. Для ограничения списка на текущих занятиях реализован фильтр «Наиболее успевающие студенты» (у которых сданы все задания в запланированные сроки) и сортировка этого списка по сумме баллов за все работы. Для определения очереди студентов для сдачи текущей работы реализован алгоритм сортировки студентов по количеству задолженностей, количеству попыток сдачи задания и студентам, которые преуспевают по заданиям. Для сдачи задолженностей на дополнительных занятиях реализован фильтр «Студенты с задолженностями», позволяющий посмотреть студентов с задолженностями одной группы. Список «должников» можно отсортировать по количеству задолженностей и количеству предпринятых попыток ликвидировать ту или иную задолженность. Также существует возможность просмотра отсортированного списка задолженностей для конкретного студента на дистанционных занятиях (первыми в списке идут те работы, которые были выданы к выполнению раньше всех).

Для выполнения анализа необходимо создать одно из 3-х занятий. Таким образом, пользователь может провести анализ студентов с задолженностями или наиболее успевающих студентов.

При создании дополнительного занятия (выборе анализа «Студенты с задолженностями») система осуществляет запрос у менеджера студентов с задолженностями, который выполняет их поиск и возвращает список. Сформированный список путем нажатия на кнопку сортируется по количеству задолженностей и количеству предпринятых попыток сдать ту работу, которая была выбрана, а затем выводится на экран.

При выборе анализа «Наиболее успевающие студенты» система осуществляет запрос у менеджера студентов, у которых все задания сданы в запланированные сроки, который выполняет их поиск и возвращает список. Сформированный список студентов сортируется по сумме баллов за все работы, а затем выводится на экран.

При создании текущего занятия (выборе анализа общего списка студентов для сдачи текущей работы) система производит сортировку по количеству задолженностей, количеству предпринятых попыток отчитать запланированную к сдаче работу (в случае, если количество задолженностей одинаково) и по количеству работ, опережающих план занятий.

При создании дистанционного занятия (выборе анализа задолженностей конкретного студента) система сортирует его задолженности по сроку сдачи, т.е. первыми будут идти те задания, у которых срок сдачи истекает раньше.

## 5.12 Формализация критериев поиска студентов с задолженностями

Для проведения анализа необходимо организовать поиск студентов, у которых есть задолженность. Студенты, у которых задолженность есть, характеризуются ненулевым значением количества задолженностей.

Пусть count\_debt – количество задолженностей у рассматриваемого студента.

В соответствии с условиями задачи:

- отсутствие задолженностей у данного студента: count\_debt = 0;

- наличие задолженностей у данного студента: count\_debt > 0.

## 5.13 Формализация критериев поиска наиболее успевающих студентов

Для проведения анализа необходимо организовать поиск студентов, у которых нет задолженностей и все работы были сданы в установленные сроки. Такие студенты характеризуются нулевым значением количества задолженностей, дата выполнения задания < срок (дата) выполнения.

Пусть count\_debt – количество задолженностей у рассматриваемого студента, tasks\_on\_time – логическое переменная, характеризующая при значении true выполнение всех выполненных работ в установленные сроки.

В соответствии с условиями задачи:

- отсутствие задолженностей у данного студента: count\_debt = 0;

- наличие свиданий у данного поклонника: count\_debt > 0;

- все выполненные работы выполнены вовремя: tasks\_on\_time = true;

- не все выполненные работы выполнены вовремя: tasks\_on\_time = false.

## 5.14 Формализация критериев сортировки студентов по количеству задолженностей, количеству предпринятых попыток сдачи и по преуспевающим программу студентам

Первоначально система производит сравнение студентов по количеству задолженностей, сортируя список в порядке возрастания. В случае совпадения количества задолженностей двух сравниваемых студентов система автоматически переходит к следующему критерию – количество попыток сдачи работы, сортируя список в порядке возрастания. Далее выбираются студенты, которые преуспевают по плану занятий (т.е. которые сдали текущую работу), и перемещаются в конец списка.

Пусть count\_debt – количество задолженностей данного студента, count\_attempts – количество предпринятых попыток сдать данную работу данного студента, task\_completed – логическая переменная, означающая при значении true, что задание выполнено.

Таким образом, в соответствии с условиями задачи, сортировка студентов по количеству задолженностей, количеству предпринятых попыток отчитать работу и количеству преуспевающих работ может производиться по следующим критериям в определенном порядке:

1. count\_debt = min;

2. count\_attemps = min.

В случае, если задание уже выполнено, студент перемещается промежуточный массив, в которым хранятся все преуспевающие студенты. После сортировки списка все студенты из промежуточного массива отправляются в конец общего отсортированного списка, а сам массив уничтожается.

## 5.15 Формализация критериев сортировки задолженностей конкретного студента

Для конкретного студента система производит сортировку его задолженностей таким образом, чтобы сроки сдачи заданий окончания были по возрастанию.

Пусть deadline – дата последняя дата сдачи задания.

В соответствии с условиями задачи:

Если (task1.deadline > task2.deadline), то отсортированный список задолженностей будет таким:

1. task2;

2. task1.

# 6 ПОРЯДОК КОНТРОЛЯ И ПРИЁМКИ СИСТЕМЫ

Испытания приложения должны проводиться в соответствии с требованиями ГОСТ 34.603-92 "Информационная технология. Виды испытаний автоматизированных систем". При реализации приложения в рамках настоящего ТЗ устанавливаются предварительные испытания на стенде Исполнителя по созданию приложения.

Приемку работ должна осуществлять приемочная комиссия, в состав которой включаются:

* представители заказчика;
* представители исполнителя.

При проведении испытаний приемочной комиссии предъявляются разработанные Исполнителем материалы (конструкторская, программная и эксплуатационная документация и программное обеспечение в исходных и исполняемых кодах). Комплектность предоставляемой документации определяется требования настоящего ТЗ.

Предварительные испытания заканчиваются подписанием приемочной комиссией протокола испытания с указанием в нем перечня необходимых доработок программного обеспечения, конструкторской, программной и эксплуатационной документации и сроков их выполнения.

После устранения замечаний, осуществляются повторные предварительные испытания приложения. На повторные предварительные испытания Исполнителем предъявляются доработанные по результатам ранее выполненных испытаний материалы. Испытания завершаются оформлением Акта готовности приложения к развертыванию в опытной зоне.

Отдельные пункты ТЗ могут изменяться и уточняться по согласованию сторон.

В недельный срок после начала работ исполнитель предоставляет на согласование заказчику план-график работ по данному этапу.

# 7 ТРЕБОВАНИЯ К СОСТАВУ И СОДЕРЖАНИЮ РАБОТ ПО ПОДГОТОВКЕ ОБЪЕКТА АВТОМАТИЗАЦИИ К ВВОДУ СИСТЕМЫ В ДЕЙСТВИЕ

В процессе создания приложения должен быть подготовлен и передан Заказчику комплект документации в составе:

* проектная документация и материалы техно-рабочего проекта на разработку приложения;
* конструкторская, программная и эксплуатационная документация на приложения;
* сопроводительная документация на поставляемые программно-аппаратные средства в комплектности поставки заводом-изготовителем;
* предложения по организации системно-технической поддержки функционирования приложения.

Состав и содержание комплекта документации на приложение может быть уточнен на стадии проектирования.

Подготовленные документы должны удовлетворять требованиям государственных стандартов и рекомендаций по оформлению, содержанию, форматированию, использованию терминов, определений и надписей, обозначений программ и программных документов.

# 8 ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ

## 8.1. Общие требования к документированию

Документы должны быть представлены на бумажном виде (оригинал) и на магнитном носителе (копия). Исходные тексты программ - только на магнитном носителе (оригинал). Возможно предоставление комплекта документации и текстов программ на компакт-дисках.

Все документы должны быть оформлены на русском языке. Состав документов на общее программное обеспечение, поставляемое в составе АИС "Приложение по учету успеваемости студентов", должен соответствовать комплекту поставки компании - изготовителя.

## 8.2. Перечень подлежащих разработке документов

В ходе создания приложения должен быть подготовлен и передан Заказчику комплект документации в составе:

* проектная документация и материалы техно-рабочего проекта на разработку приложения;
* конструкторская, программная и эксплуатационная документация на приложение;
* сопроводительная документация на поставляемые программно-аппаратные средства в комплектности поставки заводом-изготовителем;
* предложения по организации системно-технической поддержки функционирования приложения.

Состав и содержание комплекта документации на приложение может быть уточнен на стадии проектирования.

Подготовленные документы должны удовлетворять требованиям государственных стандартов и рекомендаций по оформлению, содержанию, форматированию, использованию терминов, определений и надписей, обозначений программ и программных документов.

# 9 ПОРЯДОК ВНЕСЕНИЯ ИЗМЕНЕНИЙ

Техническое задание разработано на основе следующих документов и информационных материалов:

* ГОСТ 34.003-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термин и определения.
* ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
* ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Стадии создания.
* ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
* ГОСТ 34.603-92. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем.
* ГОСТ Р 54296-2010/ISO/PAS 17002:2004 Оценка соответствия. Конфиденциальность. Принципы и требования.
* ГОСТ Р 58833-2020 Защита информации. Идентификация и аутентификация. Общие положения.
* РД 50-34.698-90. Автоматизированные системы. Требования к содержанию документов.
* ГОСТ Р 50922-2006. Защита информации.

*ПРИЛОЖЕНИЕ 3*

СОСТАВИЛИ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование организации, предприятия | Должность исполнителя | Фамилия имя, отчество | Подпись | Дата |
| Московский Политех | Исполнитель | Герасименко Д. А. |  | 17.11.2021 |
| Московский Политех | Исполнитель | Поляков В. А. |  | 17.11.2021 |
| Московский Политех | Исполнитель | Щеголькова В. В. |  | 17.11.2021 |

СОГЛАСОВАНО

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование организации, предприятия | Должность исполнителя | Фамилия имя, отчество | Подпись | Дата |
| Московский Политех | Заказчик | Красников А. С. |  |  |