Министерство образования и науки Российской Федерации

ФГБОУ ВО «СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

НАУКИ И ТЕХНОЛОГИЙ ИМЕНИ АКАДЕМИКА М.Ф.РЕШЕТНЕВА»

(СибГУ имени М.Ф.Решетнева)

Институт информатики и телекоммуникаций

Кафедра информационно-управляющих систем

Направление: 09.04.04 «Программная инженерия»

Направленность: Системы программной поддержки жизненного цикла изделий

**ЛАБОРАТОРНАЯ РАБОТА**

**по дисциплине «**Разработать регламент компании-разработчика ПО**»**

**Тема:** «Разработать

регламент сопровождения системы»»

(ИУС.00.00.00.027 ПЗ)

Выполнил:

Студент 1-го курса группы МПЦ23-01

Д.В. Лунев

(подпись, дата сдачи)

Проверил:

Н.С. Черниченко

(подпись, дата сдачи)

Работа защищена с оценкой

Красноярск, 2024

**Ход работы**

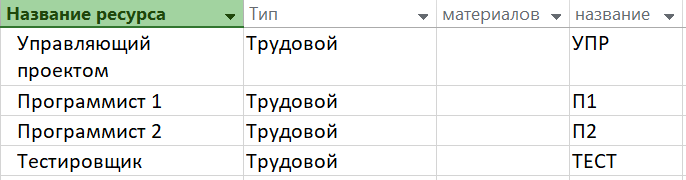
1. **Введение**

Настоящий Регламент устанавливает порядок и правила выполнения работ по созданию системы управления бассейном.

1. **Наименование процесса вышестоящего уровня**

Создание программного продукта.

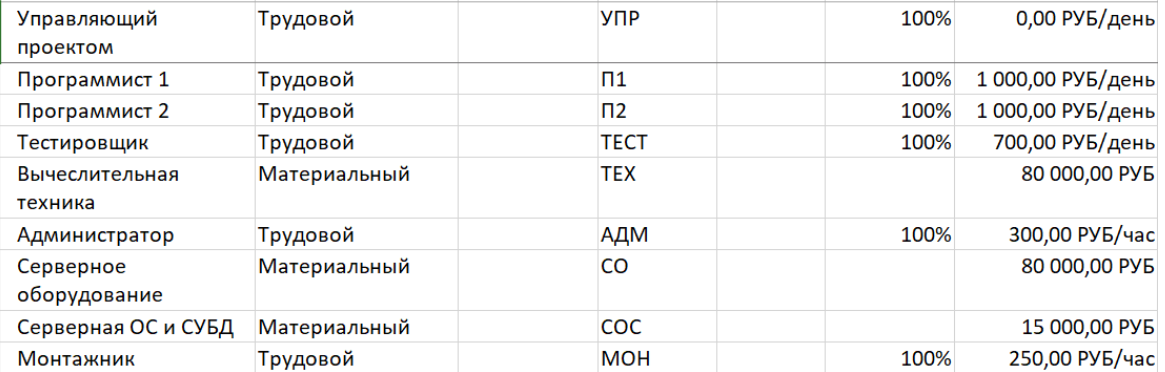
1. **Ответственный за выполнение процесса, участники процесса**



1. **Нормативные документы:**
   1. **Техническое задание**
   2. **Частные технические задания**
   3. **Сценарий использования**
   4. **Сценарий тестирования**
2. **Плановые документы:**
   1. **План работ**
   2. **Техническая документация**
   3. **Календарный план проекта**
3. **Отчетные документы по процессу:**
   1. Отчёты о ходе работ
   2. Протоколы
   3. Отчёты о результатах активностей
   4. Журналы
4. **Показатели процесса**

|  |  |  |
| --- | --- | --- |
| **Ключевые измерения** | **Приемлемый уровень** | **Пояснение** |
| Устойчивость к утечкам памяти | Система полностью очищает операционную память от неиспользуемых ею данных | Система не должна вызывать отказы и замедление своей работы из-за переполнения операционной памяти |
| Восстановление после отказов | Программа может восстановиться после критической ошибки с промежуточным результатом работы | Программа сохраняет промежуточные результаты работы и способна определить и восстановить последнее состояние перед критической ошибкой. |
| Адаптивность серверного приложения | Серверная часть программы не привязана к конкретной операционной системе. | Программа не должна содержать жесткой привязки к определенной операционной системе на случай миграции серверов. |
| Адаптивность к современным СУБД | Программа может работать с СУБД Microsoft SQL, MySQL, PostgreSQL | Программа должна быть предусмотрена интеграция с несколькими современными реляционными СУБД на случай миграции. |
| Удобство навигации клиентского интерфейса | Функции разбиты на очевидные категории | Функциональные элементы интерфейса и их окна должны быть сгруппированы в категории, состав и название которых должно быть наиболее очевидно пользователям |
| Модульность программы | Программа поделена на независимые функциональные модули | Программа должна быть поделена на модули, которые могут общаться друг с другом, но их работы не зависит друг от друга напрямую. Работа всей системы не будет остановлена из-за ошибки в одном из модулей. |
| Обслуживание модулей системы | Модули программы можно заменить, без остановки всей системы | При обслуживании и обновлении модулей системы, ход работы всей системы не останавливается. |
| Встроенная документация в клиентские приложения | Всплывающие подсказки и ссылки на разделы справки | Чтобы сократить недопонимания по использованию пользовательских приложений и интерфейсов в программе должен быть встроены справочный материал и подсказки к элементам |
| Безопасность передачи данных по сети | Используются протоколы TLS и HTTPS | Передача данных между системой и связанными приложениями и программами должно производится с использованием современных стандартов TLS и HTTPS. |
| Адаптируемый клиент | Клиентское приложение открывается на стационарном компьютере/ноутбуке или в виде web-интерфейса | Клиентское пользовательское приложение реализовано под десктопное приложение (рабочее место) и web-интерфейс (доступ через браузер, удаленный/мобильный доступ) |
| Масштабируемый обмен данными по сети | Обмен данными по протоколу http и архитектуре REST | Программа должна быть способна передавать и получать данные с большого количества источников в сети Интернет. Для обеспечения масштабируемости обмена стоит использовать архитектуру REST. |
| Производительность | С увеличением числа работников, заявок, материалов в базе в пределах нормы не должно увеличиваться время обработки заявок(10000) | Обработка заявки должна производиться быстрее чем при ручном методе. |
| Масштабируемость | Увеличение вычислительных ресурсов должно сокращать время расчётов линейно | Расчеты должны выполняться не больше чем за установленное время. Для этого с увеличением вычислительных ресурсов время расчетов должно уменьшаться хотя бы линейно |
| Отказоустойчивость | Система продолжает работать при отказе модуля | Система должна обеспечивать бесперебойную работу в течение длительного периода времени, даже при отказе отдельных модулей |
| Устойчивость к неправильному вводу данных | Программа способна определить неправильный ввод от пользователя и сообщить | Программа должна уметь определять некорректно введенные данные от пользователя и предупреждать об этом |

1. **Границы процесса:**
   1. В процесс не входит техническая поддержка пользователей
   2. В процесс не входит полное изменение концепции проекта на любом из этапов разработки
   3. В процесс не входит продажа проекта в онлайн сервисах
   4. В процесс не входит обновление проекта
   5. В процесс не входит добавление новых платформ для проекта
2. **Требования к ресурсам**

****

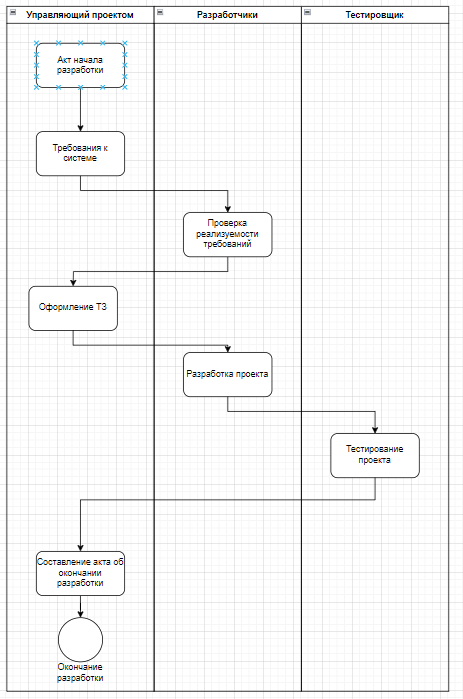
1. **Длительность выполнения процесса:**

**225 дней.**

1. **Периодичность выполнения процесса:**

**1 на каждую разработку.**

1. **Графическая модель процесса:**

****

**Github: https://github.com/JaRusskii/Busss**