

Функционально-технические задачи на разработку проекта

Учёт участия студентов в олимпиадах и конкурсах

Содержание

1	Список терминов и сокращений	2
2	Цели и задачи проекта	2
3	Роли пользователей и их полномочия	3
3.1	Пользователи и бизнес-роли	3
3.2	Таблица и матрица соответствия ролей	3
4	Функционально-технические требования	3
5	Технологические требования	5
5.1	Бэкенд	5
5.2	Фронтенд	5
5.3	СУБД	6
5.4	Совместимость и версии	6

1 Список терминов и сокращений

Таблица 1. Список принятых терминов и сокращений

Сокращение / Термин	Определение
Реестр участия	Совокупность записей об участии студентов в соревнованиях, включая конкурс, уровень, результат, год и связанного преподавателя/подразделение при необходимости.
Студент	Обучающийся образовательной организации; в системе хранится ФИО, номер студенческого билета, факультет/институт, при необходимости — кафедра и группа.
Конкурс (олимпиада)	Научное или профессиональное соревнование с названием, организатором (опционально) и возможной привязкой к направлению подготовки.
Уровень	Категория уровня соревнований: вузовский, региональный, всероссийский (при необходимости расширяемый перечень).
Результат	Итог участия: победитель, призёр, участник и т. п.
Архив/блокировка	Состояние записей или периода, когда редактирование запрещено по регламенту (например, по окончании срока подачи отчётов).
База данных	Совокупность данных, организованная для хранения и обработки в ЭВМ.
Реляционная база данных	База данных, основанная на реляционной модели данных.
Нормальная форма	Свойство отношения, характеризующее степень избыточности данных; для проекта используется нормализация до третьей нормальной формы.
Серверная часть системы	Программа, выполняющая запросы от клиента.
Клиентская часть системы	Программа, с помощью которой пользователь инициирует запросы к серверной части.

2 Цели и задачи проекта

Таблица 2. Цели и задачи проекта

№	Цели	Задачи
1	Автоматизация учёта участия студентов в олимпиадах/конкурсах	Разработка модульной системы ведения, просмотра, поиска и отчётности по участиям
2	Обеспечение актуальности и прозрачности данных	Исключение дублей, блокировка редактирования по регламенту, отслеживание изменений

3 Роли пользователей и их полномочия

3.1 Пользователи и бизнес-роли

Функционал ролей описан только в части функций, реализуемых в рамках проекта.

Таблица 3. Концепция ролей и полномочий

Бизнес-роль	Основной функционал роли
Система	Обеспечение процесса автоматизации учёта участия студентов
Администратор системы	Сотрудник, выполняющий настройку Системы на сервере, и обеспечивающий её работоспособность
Пользователь	Сотрудник, управляющий реестром участия, поиском и просмотром достижений

3.2 Таблица и матрица соответствия ролей

Матрица отражает соответствие бизнес-роли и системной роли для разрабатываемой системы.

Таблица 4. Описание матрицы «Бизнес-роли / Системные роли»

Бизнес-роль	Системная роль (код)
Система	Система (system)
Администратор системы	Администратор (admin)
Пользователь	Пользователь (user)

4 Функционально-технические требования

Таблица 5. Функционально-технические требования к разрабатываемой системе

№	Название функции	Описание функции	Бизнес-роль
ФТ-1	Хранение данных	<ul style="list-style-type: none"> Система должна обеспечить хранение всех данных в реляционной базе данных. Система должна хранить все данные в нормализованном виде не ниже третьей нормальной формы. Система должна хранить следующие данные: <ul style="list-style-type: none"> Студенты: ФИО, номер студенческого билета, факультет/институт, (опционально) кафедра, группа. Конкурсы: наименование, организатор (опционально), тип (олимпиада/конкурс/чемпионат и т. п.). Уровни: кодификатор уровней участия (вузовский/региональный/всероссийский/...). Результаты: кодификатор итогов (победитель/призёр/участник/...). Участия: ссылка на студента, конкурс, уровень, результат, год участия, (опционально) научный руководитель/куратор, комментарий, признак блокировки редактирования. 	Система
ФТ-2	Размещение системы	<ul style="list-style-type: none"> Система должна быть разделена на следующие функциональные модули: <ul style="list-style-type: none"> База данных; Серверная часть системы; Клиентская часть системы. Разрабатываемая система должна быть независима от серверов размещения функциональных модулей. 	Система
ФТ-3	Настройка системы через конфигурационные файлы	<ul style="list-style-type: none"> Система должна предоставлять Администратору функциональную возможность настройки серверной и клиентской частей, в части указания необходимых параметров конфигурационных файлов для межмодульной связи. 	Администратор

№	Название функции	Описание функции	Бизнес-роль
ФТ-4	Ведение реестра участников	<ul style="list-style-type: none"> • ФТ-4.1. Добавление участия. Система должна предоставлять функциональную возможность создания записи об участии студента. Создание дубликата записи (одинаковые студент, конкурс и год участия) должно быть недоступно. • ФТ-4.2. Просмотр достижений. Система должна предоставлять функциональную возможность просмотра достижений: <ul style="list-style-type: none"> – по студенту — список участия с группировкой по годам и указанием конкурса, уровня и результата; – по конкурсу — список участников и их результатов с возможностью фильтрации по году и уровню. • ФТ-4.3. Поиск и фильтрация. Система должна предоставлять функциональную возможность поиска и фильтрации по году участия (диапазон/точный), по уровню (из кодификатора), по факультету/институту студента; допускается частичный поиск по ФИО и названию конкурса. • ФТ-4.4. Блокировка редактирования по регламенту. Система должна поддерживать блокировку редактирования записей участников по окончании срока подачи отчётов (регламентного периода). 	Пользователь

5 Технологические требования

5.1 Бэкенд

Серверная часть должна быть реализована на **одной** из следующих технологий (на выбор команды): **Java, Kotlin, Go**.

- Архитектура API: REST/JSON; кодировка UTF-8.
- Сборка и запуск: стандартные инструменты экосистемы выбранного языка (например, Maven/Gradle для Java/Kotlin; стандартная toolchain для Go).

5.2 Фронтенд

Клиентская часть должна быть реализована на **React** или **Vue** (на выбор команды).

- Сборка: Node.js для сборки (версия указывается в документации проекта).
- Маршрутизация/состояние: библиотека на выбор команды.
- Взаимодействие с сервером: REST/JSON; поддержка CORS.

5.3 СУБД

Обязательная система управления базами данных: **PostgreSQL**.

- Схема данных в нормальной форме не ниже третьей нормальной формы (3НФ).
- Параметры подключения и процедура инициализации базы данных должны быть задокументированы.
- Требования к резервному копированию описаны в эксплуатационной документации проекта.

5.4 Совместимость и версии

- Конкретные версии используемых технологий фиксируются в документации проекта (например, в **README .md**).
- Использование технологий вне указанного перечня допускается только по согласованию с преподавателем.