

Каранда Алексей Юрьевич

студент, Поволжский государственный университет телекоммуникаций и информатики, Россия, г. Самара

Куваева Евгения Николаевна

заместитель декана по учебно-воспитательной работе, старший преподаватель кафедры информационных систем и технологий, Поволжский государственный университет телекоммуникаций и информатики, Россия, г. Самара

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЁТА И АНАЛИЗА УСПЕВАЕМОСТИ СТУДЕНТОВ

Аннотация. Статья посвящена задаче проектирования информационной системы (ИС) учёта успеваемости студентов вуза. Рассмотрены существующие способы ведения такого учёта: табличные редакторы, модули систем дистанционного обучения и корпоративные ИС вуза, выделены их основные недостатки. Предложена концепция ИС с клиент-серверной архитектурой и реляционной базой данных, включающая модули ввода оценок, формирования ведомостей и построения отчётов по успеваемости. Описана структура основных таблиц базы данных, представлены пользовательские сценарии для преподавателя, сотрудника деканата и студента. Предложенное решение может быть использовано как основа для разработки прототипа системы в рамках учебной практики.

Annotation. The article is devoted to the design of an information system (IS) for recording student academic performance at a university. Existing methods of maintaining such records are examined - spreadsheet editors, distance-learning system modules, and corporate university information systems - and their main drawbacks are identified. A concept of an IS with a client-server architecture and a relational database is proposed, including modules for entering grades, generating grade sheets, and building performance reports. The structure of the main database tables is described, and user scenarios are presented for a lecturer, a dean's office employee and a student.

The proposed solution can be used as a basis for developing a system prototype within academic practice.

Ключевые слова: информационная система, учёт успеваемости, база данных, ER-модель, клиент-серверная архитектура, автоматизация образовательного процесса, use-case.

Keywords: information system, academic performance tracking, database, ER model, client-server architecture, automation of the educational process, use case.

1. Введение

Одной из самых трудоёмких задач в работе вуза остаётся учёт успеваемости студентов: фиксация оценок за контрольные работы, зачёты и экзамены, формирование ведомостей, подготовка отчётов для деканата и кафедры. Во многих вузах, особенно небольших, эта работа до сих пор ведётся вручную либо в обычных табличных редакторах, которые для такой задачи подходят плохо [1, 7].

Актуальность темы связана с тем, что объём данных об учебном процессе постоянно растёт: становится больше дисциплин, форм контроля и студентов. Кроме того, вузу нужно вовремя выявлять студентов с задолженностями и отслеживать динамику успеваемости, а таблицы и модули LMS для этого приспособлены слабо: они плохо связаны между собой, из-за чего данные дублируются и расходятся. Проблема в том, что на уровне кафедры или факультета обычно нет отдельного удобного инструмента для хранения данных об успеваемости с разграничением доступа между преподавателями, деканатом и студентами и с возможностью автоматически строить отчёты.

Цель работы - спроектировать концепцию информационной системы учёта и анализа успеваемости студентов: определить архитектуру, модель данных и основные сценарии работы пользователей, чтобы в дальнейшем на этой основе можно было реализовать систему. Для этого решались следующие задачи:

- изучить предметную область и посмотреть, какие средства учёта успеваемости уже используются;
- определить требования к проектируемой системе;

- разработать модель данных;
- предложить архитектуру и технологии реализации;
- описать основные сценарии взаимодействия пользователей с системой.

2. Анализ предметной области и существующих решений

2.1 Предметная область

Учёт успеваемости в вузе связан с несколькими процессами: формированием учебного плана, распределением нагрузки преподавателей, проведением текущего контроля и промежуточной аттестации, формированием ведомостей, расчётом среднего балла и других показателей. Основные сущности здесь - студент, преподаватель, дисциплина, группа, форма контроля и оценка. Эти сущности довольно жёстко связаны между собой, поэтому предметную область удобно представить в виде реляционной модели данных [4, 5].

2.2 Обзор существующих решений

Можно выделить три группы решений, которые сейчас применяются для учёта успеваемости [2].

– табличные редакторы (Excel, Google Таблицы) - самый распространённый вариант на уровне отдельного преподавателя или кафедры. Плюсы: простота и отсутствие затрат на внедрение. Минусы: нет единого хранилища данных, при работе нескольких человек с разными копиями файла данные быстро расходятся, нет разграничения доступа, сложно строить сводную аналитику;

– модули LMS (Moodle, «Открытое образование» и похожие платформы) - хорошо фиксируют результаты тестов и заданий внутри электронного курса, но изначально созданы для другого - для поддержки учебного процесса, а не для официальной отчётности по успеваемости. С формированием ведомостей и сводной аналитикой деканата такие системы связаны слабо;

– корпоративные ИС вуза (1С:Университет, «Галактика Управление вузом» и подобные) - функционально самые полные, но это тяжёлые и дорогие системы, рассчитанные на весь вуз целиком. Для локальной задачи одной

кафедры или факультета они избыточны, а использовать их как учебный проект практически невозможно.

Получается, что между простыми табличными инструментами и громоздкими корпоративными ИС есть разрыв: специализированного, но не слишком сложного решения на рынке фактически нет. Поэтому разработка собственной концепции ИС под нужды конкретного подразделения выглядит оправданной. Сравнительный анализ представлен в таблице 1.

Таблица 1 - Сравнительный анализ аналогов информационной системы учёта успеваемости

Категория решения	Преимущества	Недостатки / ограничения	Применимость на уровне кафедры
Табличные редакторы (Excel, Google Sheets)	Низкий порог входа, простота, без затрат на внедрение	Нет централизации данных, слабый контроль доступа, риск рассинхронизации	Высокая, но с рисками ошибок
Модули LMS (Moodle и др.)	Автоматизация тестов, интеграция с курсами	Слабая связь с сведениями деканата, нет сквозной аналитики	Средняя
Корпоративные ИС (1С:Университет и др.)	Полнота функций, интеграция в масштабе вуза	Высокая стоимость, сложная настройка, тяжеловесность	Низкая (избыточная сложность)
Проектируемая ИС (предлагаемое решение)	Специализация, лёгкость, гибкая аналитика	Требует разработки и внедрения	Высокая (оптимальный баланс)

2.3 Требования к проектируемой системе

Исходя из анализа предметной области и потребностей пользователей (преподавателей, деканата/кафедры, студентов), можно сформулировать следующие требования.

– функциональные: ведение справочников студентов, преподавателей и дисциплин; ввод и правка оценок текущего контроля и промежуточной аттестации; автоматическое формирование ведомостей; построение отчётов (средний балл по группе или дисциплине, доля студентов с задолженностью, динамика успеваемости по семестрам);

– нефункциональные: разграничение прав по ролям, целостность данных, приемлемая скорость формирования отчётов, возможность подключения новых кафедр.

3. Описание предлагаемого решения

3.1 Архитектура системы

Предлагается клиент-серверная архитектура из трёх уровней: уровень представления (веб-интерфейс), уровень бизнес-логики (серверное приложение) и уровень хранения данных (реляционная СУБД). Такое разделение позволяет менять интерфейс, не трогая логику обработки данных, и облегчает дальнейшую поддержку системы. Архитектура системы представлена на рисунке 1.



Рисунок 1 - Архитектура системы

Уровень представления выполнен в виде веб-приложения - это удобно, потому что не нужно ничего устанавливать на компьютеры пользователей, и работать с системой можно с любого устройства с браузером. Уровень бизнес-логики отвечает за обработку запросов, проверку прав доступа, проверку корректности вводимых оценок и формирование отчётов. Уровень хранения данных - это реляционная СУБД, где хранятся данные обо всех сущностях предметной области с сохранением связей между ними.

3.2 Технологический стек

Для реализации предлагаемой концепции выбраны следующие технологии:

- серверная часть - Python (Flask/Django) либо PHP для реализации REST-подобного взаимодействия между клиентом и сервером;
- СУБД - PostgreSQL или MySQL, зрелые бесплатные реляционные базы данных с поддержкой транзакций и внешних ключей;
- клиентская часть - HTML, CSS, JavaScript, а также библиотека Chart.js для построения графиков успеваемости;
- отчётность - диаграммы на стороне клиента и экспорт данных в форматы офисных пакетов.

Такой стек выбран потому, что все компоненты бесплатны, широко распространены и хорошо документированы, а значит, систему проще разрабатывать и в дальнейшем поддерживать силами штатных специалистов вуза.

3.3 Модель данных

В основе модели данных лежат сущности «Студент», «Преподаватель», «Дисциплина», «Группа», «Учебный план», «Оценка», «Форма контроля». Между «Студентом» и «Дисциплиной» связь «многие ко многим», реализованная через сущность «Оценка», которая также связана с «Преподавателем» (кто поставил оценку) и «Формой контроля» (в рамках какого мероприятия). Такое построение модели соответствует принципам нормализации реляционных баз данных и позволяет избежать дублирования информации [6]. На рисунке 2 представлена ER-диаграмма информационной системы учёта успеваемости. В таблице 2 представлена структура основных таблиц базы данных.

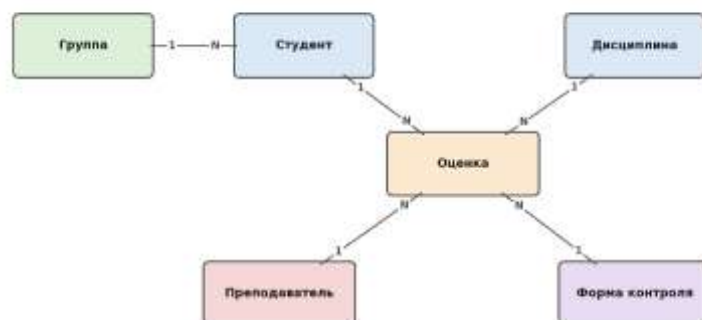


Рисунок 2 - ER-диаграмма информационной системы учёта успеваемости

Таблица 2 - Структура основных таблиц базы данных

Таблица	Поле (PK/FK)	Тип данных	Описание
Студент (Student)	ID_студента (PK)	INT	Уникальный идентификатор
	ФИО	VARCHAR(150)	Полное имя студента
	ID_группы (FK)	INT	Ссылка на учебную группу
	Дата_рождения	DATE	Дата рождения
	Статус	VARCHAR(50)	Обучается / отчислен / академ. отпуск
Дисциплина (Subject)	ID_дисциплины (PK)	INT	Уникальный идентификатор
	Наименование	VARCHAR(150)	Название учебного предмета
	Семестр	INT	Номер семестра
	Часы	INT	Объём академических часов
	Кафедра	VARCHAR(100)	Закреплённая кафедра
Преподаватель (Teacher)	ID_преподавателя (PK)	INT	Уникальный идентификатор
	ФИО	VARCHAR(150)	Полное имя преподавателя
	Должность	VARCHAR(100)	Должность / учёное звание
	Кафедра	VARCHAR(100)	Кафедра
Оценка (Grade)	ID_оценки (PK)	INT	Первичный ключ

Таблица	Поле (PK/FK)	Тип данных	Описание
	ID_студента (FK)	INT	Ссылка на студента
	ID_дисциплины (FK)	INT	Ссылка на дисциплину
	ID_преподавателя (FK)	INT	Ссылка на экзаменатора
	Форма_контроля	VARCHAR(50)	Экзамен / зачёт / КР / лаб.
	Значение	INT VARCHAR(20)	Оценка (2-5 или «зачтено»)
	Дата	DATE	Дата выставления
Группа (Group)	ID_группы (PK)	INT	Первичный ключ
	Шифр_группы	VARCHAR(20)	Обозначение группы (напр. ИВТ-31)
	Курс	INT	Номер курса
	Факультет	VARCHAR(100)	Название факультета / института

3.4 Пользовательские сценарии (use-case)

Выделены три категории пользователей системы: преподаватель, сотрудник деканата/кафедры и студент. Для каждой категории определён свой набор сценариев работы с системой. На рисунке 3 представлена диаграмма прецедентов.

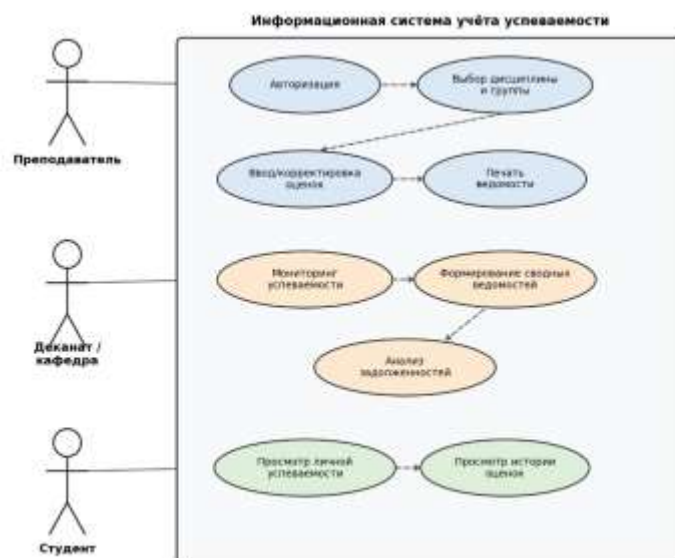


Рисунок 3 - Диаграмма прецедентов (use-case)

Основные сценарии:

- преподаватель входит в систему, выбирает закреплённую дисциплину и группу, вносит оценки текущего и промежуточного контроля;
- сотрудник деканата формирует сводную ведомость успеваемости по группе или потоку за выбранный период, получает отчёт по числу студентов с академической задолженностью;
- студент входит в систему в режиме просмотра и видит свои результаты по всем изучаемым дисциплинам.

3.5 Расчёт аналитических показателей

Система рассчитывает несколько показателей: средний балл по дисциплине или группе, процент студентов, успешно прошедших аттестацию, процент студентов с задолженностью. Отчёт строится в три шага: сначала выбираются записи об оценках за период по заданным критериям (группа, дисциплина, семестр), затем данные агрегируются (считаются среднее значение и доли), после чего результат отображается в виде графиков в клиентском приложении [3]. На рисунке 4 представлен прототип интерфейса приложения.

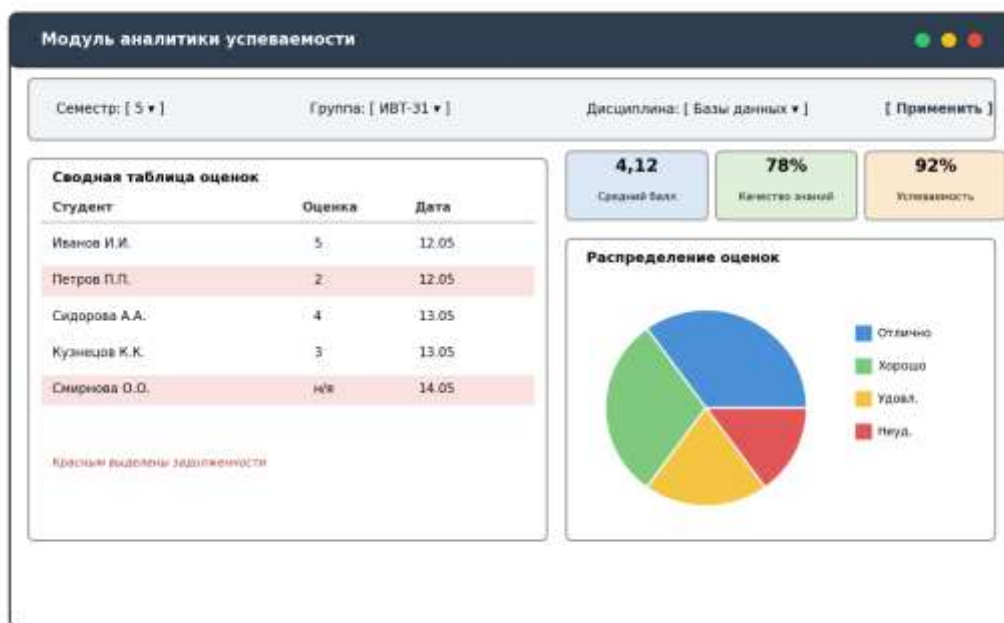


Рисунок 4 - Прототип интерфейса модуля аналитики успеваемости

4. Заключение

В статье проанализирована предметная область учёта успеваемости и рассмотрены существующие подходы к её автоматизации. Показано, что между простыми табличными решениями и тяжёлыми корпоративными ИС вуза есть разрыв, который можно закрыть отдельной специализированной системой. Предложена концепция такой системы: трёхуровневая архитектура, реляционная модель данных, сценарии для трёх категорий пользователей и технологический стек на основе свободно распространяемого программного обеспечения.

Практическая значимость работы состоит в том, что предложенную концепцию можно использовать как основу для разработки прототипа системы в рамках учебной или производственной практики, а затем - для внедрения на уровне отдельной кафедры или факультета без больших затрат, характерных для корпоративных ИС масштаба вуза.

В дальнейшем работу можно продолжить в направлении интеграции с другими университетскими системами (расписание, электронное портфолио студента), добавления прогнозирования риска академической задолженности и проработки вопросов защиты персональных данных студентов в соответствии с законодательством Российской Федерации.

Литература

1. Грекул, В. И. Проектное управление в сфере информационных технологий [Электронный ресурс] / Н. Л. Коровкина, Ю. В. Куприянов, В. И. Грекул. - 3-е изд. (эл.). - М. : Лаборатория знаний, 2020. - 339 с. - Режим доступа: <https://lib.rucont.ru/efd/443585>.
2. Секлетова, Н. Н. Анализ рынка информационных систем и технологий [Электронный ресурс] : учеб. пособие / Н. Н. Секлетова, А. С. Тучкова, О. И. Захарова ; ПГУТИ, Каф. ИСТ. - Самара : ПГУТИ, 2018. - 215 с.
3. Круз, Р. Л. Структуры данных и проектирование программ [Электронный ресурс] / пер.: К. Г. Финогенов, Р. Л. Круз. - 3-е изд. (эл.). - М. : Лаборатория знаний, 2017. - 768 с.
4. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг. - 3-е изд. - М. : Вильямс, 2019. - 1440 с. - ISBN 978-5-8459-0788-7.
5. Дейт, К. Дж. Введение в системы баз данных / К. Дж. Дейт. - 8-е изд. - М. : Вильямс, 2021. - 1328 с. - ISBN 978-5-8459-1051-1.
6. Голицына, О. Л. Базы данных : учебное пособие / О. Л. Голицына, Н. В. Максимов, И. И. Попов. - 4-е изд., перераб. и доп. - М. : Форум, ИНФРА-М, 2022. - 400 с. - ISBN 978-5-00091-828-3.
7. Царев, Р. Ю. Проектирование информационных систем : учебное пособие / Р. Ю. Царев, А. Н. Пупков. - Красноярск : СФУ, 2019. - 224 с.

Literature

1. Grekul, V. I. Project management in the field of information technology [Electronic resource] / N. L. Korovkina, Yu. V. Kupriyanov, V. I. Grekul. — 3rd ed. (electronic). — Moscow : Laboratoriya Znaniy, 2020. — 339 p. — Access mode: <https://lib.rucont.ru/efd/443585>.
2. Sekletova, N. N. Analysis of the information systems and technologies market [Electronic resource] : study guide / N. N. Sekletova, A. S. Tuchkova, O. I. Zakharova ; PSUTI, Dept. of IST. — Samara : PSUTI, 2018. — 215 p.

3. Kruse, R. L. Data structures and program design [Electronic resource] / transl. by K. G. Finogenov. — 3rd ed. (electronic). — Moscow : Laboratoriya Znaniy, 2017. — 768 p.
4. Connolly, T. Database systems: a practical approach to design, implementation, and management / T. Connolly, C. Begg. — 3rd ed. — Moscow : Williams, 2019. — 1440 p. — ISBN 978-5-8459-0788-7.
5. Date, C. J. An introduction to database systems / C. J. Date. — 8th ed. — Moscow : Williams, 2021. — 1328 p. — ISBN 978-5-8459-1051-1.
6. Golitsyna, O. L. Databases : study guide / O. L. Golitsyna, N. V. Maksimov, I. I. Popov. — 4th ed., rev. and enl. — Moscow : Forum, INFRA-M, 2022. — 400 p. — ISBN 978-5-00091-828-3.
7. Tsarev, R. Yu. Information systems design : study guide / R. Yu. Tsarev, A. N. Pupkov. — Krasnoyarsk : SFU, 2019. — 224 p.