

УДК 004.4

Гребнева Д. М.

к.п.н., доцент кафедры ИТ

Филиал РГППУ в г.Нижний Тагил

г. Нижний Тагил, Россия

ПРОЕКТИРОВАНИЕ СИСТЕМЫ СОЗДАНИЯ ТЕСТОВ ДЛЯ ШКОЛЫ ЮНОГО ИНЖЕНЕРА «ЭКСПОНЕНТА»

Аннотация

В статье описан процесс проектирования системы создания тестов на примере школы юного инженера «Экспонента» для решения проблемы систематического учета и централизованного хранения данных об успеваемости обучающихся. Цель статьи – предложить проект системы создания тестов в нотации UML. Концептуальная модель веб-приложения представлена в виде диаграммы вариантов использования. Обозначенные варианты использования описаны и детализированы в форме диаграмм последовательности. Представленные в статье материалы могут быть интересны студентам, изучающим проектирование и разработку веб-приложений, и использованы как основа для реализации подобного рода систем создания тестов.

Ключевые слова: система для создания тестов, проектирование, объектно-ориентированный подход.

Grebneva D. M.

Ph.D., Associate Professor of the IT Department

Branch of the RGPPU in Nizhny Tagil

Nizhny Tagil, Russia

THE PROCESS OF DESIGN THE TEST CREATION SYSTEM FOR THE SCHOOL OF YOUNG ENGINEER «EXPONENT»

Abstract

The article describes the process of designing a system for creating tests on the example of the school of a young engineer "Exhibitor" to solve the problem of systematic accounting and centralized storage of data on student progress. The purpose of the article is to propose a project of a system for creating tests in UML notation. The conceptual model of a web application is presented as a use case

diagram. The identified use cases are described and detailed in the form of sequence diagrams. The materials presented in the article may be of interest to students studying design and development of web applications, and can be used as a basis for the implementation of this kind of test creation systems.

Key words: system for creating tests, design, object-oriented approach.

В настоящее время все больше разнообразных творческих объединений, кружков, школ развития работают в дистанционной или смешанной форме. При этом важно систематически оценивать и вести учет результатов тестов, контрольных, творческих и проектных работ обучающихся [2]. Исходя из обозначенной проблемы, актуальным становится разработка системы создания тестов для оценивания знаний обучающихся на примере работы школы юного инженера «Экспонента».

Школа юного инженера «Экспонента» функционирует на базе Нижнетагильского государственного социально-педагогического института (филиала) федерального государственного автономного образовательного учреждения высшего образования «Российского государственного профессионально-педагогического университета». Основная деятельность школы направлена на дополнительное обучение современным направлениям в информатике и ИКТ обучающихся в возрасте от 9 до 17 лет. В рамках этой школы систематически проводятся обучающие мероприятия и возникает проблема автоматизации проверки полученных знаний обучающихся. В качестве решения данной проблемы можно предложить создание клиент-серверной системы создания тестов для обучающихся, которая позволит создавать необходимые тесты, централизованно хранить данные и формировать необходимые отчеты [3].

Для построения концептуальной модели системы онлайн-мониторинга используем объектно-ориентированный подход. Описать предметную область в терминах объектно-ориентированного подхода можно на универсальном языке моделирования UML. Выделяют следующие типы диаграмм: диаграмма вариантов использования (для описания функциональных требований к системе), диаграмма последовательностей и диаграмма деятельности прецедента.

Диаграмма вариантов использования для системы создания тестов представлена на рисунке 1. Вариант использования представляет собой последовательность действий (транзакций), осуществляемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим

лицом). Вариант использования изображает типичное взаимодействие между пользователем и системой.

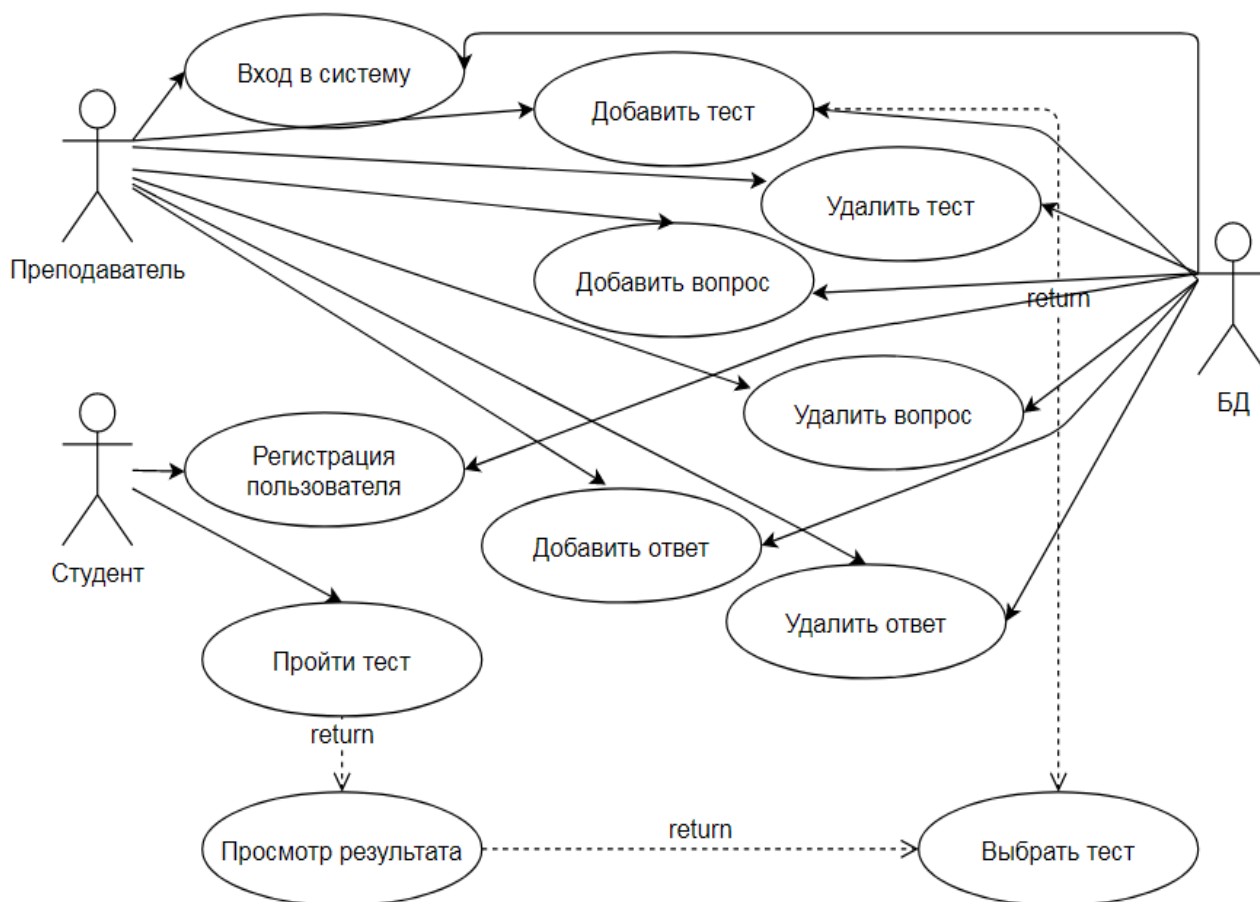


Рис. 1. Диаграмма вариантов использования

В результате анализа предметной области нами были выделены действующие лица – преподаватель, студент, база данных (БД) и варианты использования – вход в систему, регистрация пользователей, создание тестов, прохождение тестов, просмотр результатов. Таким образом, основные функции системы для создания тестов – это предоставление возможности преподавателю создавать свои тесты, предоставление пользователю удобного интерфейса для прохождения теста, централизованное хранение данных.

После разработки диаграммы прецедентов и выявления действующих лиц приступают к точному определению каждого прецедента. Этот процесс называется детализацией прецедентов [1]. Как итог этой деятельности будет получен более детализированный прецедент, включающий, как правило, имя прецедента и его спецификацию. Спецификации прецедентов представлены для прецедентов: добавить тест, добавить вопрос, пройти тест (таблицы 1-3).

Таблица 1

Спецификация прецедента «Добавить тест»

Прецедент	Добавить тест.
ID	1
Краткое описание	Добавление нового теста.
Главное действующее лицо	Преподаватель
Предусловие:	Прецедент начинается, когда Преподаватель выбрал опцию «Добавить тест»
Основной поток	<ol style="list-style-type: none"> 1. Система запрашивает у Преподавателя название теста. 2. Если тест с данным именем уже существует в базе данных <ol style="list-style-type: none"> 2.1. Выдать сообщение об ошибке 3. Иначе <ol style="list-style-type: none"> 3.1. Система добавляет новый тест. 4. Обновленный список тестов сохраняется в базе данных.
Постусловия	Система обновляет список тестов.

Таблица 2

Спецификация прецедента «Добавить вопрос»

Прецедент:	Добавить вопрос.
ID	2
Краткое описание	Добавление вопросов в тест.
Главное действующее лицо	Преподаватель.
Предусловие	Прецедент начинается, когда Преподаватель выбрал тест, а также выбрал опцию «Добавить вопрос».
Основной поток	<ol style="list-style-type: none"> 1. Система запрашивает у Преподавателя ввод данных. 2. Преподаватель вводит содержание вопроса, варианты ответов и отмечает правильный ответ на поставленный вопрос. 3. Преподаватель подтверждает добавление нового вопроса. 4. Система проверяет корректность введенных данных, а именно наличие ответов, содержание вопроса и правильный ответ. 5. Если проверка данных не проведена успешно <ol style="list-style-type: none"> 5.1. Выдать сообщение о некорректных данных. 6. Иначе <ol style="list-style-type: none"> 6.1. Система добавляет новый вопрос. 7. Вопрос сохраняется в базе данных.
Постусловия	Система обновляет список вопросов на экране.

Таблица 3

Спецификация прецедента «Пройти тест»

Прецедент	Пройти тест.
ID	3
Краткое описание:	Прохождение тестирования Студентом
Главные актеры:	Студент.
Предусловие:	Прецедент начинается, когда Студент начал прохождение теста
Основной поток:	<ol style="list-style-type: none"> 1. Система отображает список доступных Студенту тестов. 2. Студент выбирает тест и выбирает опцию «выполнить». 3. Система инициирует начало тестирования и выдает Студенту вопросы, пока все вопросы не были заданы. 4. Студент должен ответить на текущий вопрос, чтобы пройти к следующему. 5. По прохождении теста система выдает на экран результаты тестирования .
Постусловия:	1. Система фиксирует результат выполнения тестирования в базу данных.
Альтернативные потоки:	Выполнение теста прервано.

Для наглядного изображения поведения прецедентов удобно применять диаграмму деятельности. Диаграмма деятельности похожа на блок-схемы, с помощью нее можно планировать переходы по условию и завершающие действия, которые происходят при благоприятных исходах всех условий. Так на диаграмме деятельности «Добавить вопрос» (рис. 2) проверяется выполнение трех условий: заполнение поля «вопрос» (поле не пустое), добавление ответов (добавление нескольких ответов на вопрос) и наличие правильного ответа (есть хотя бы один правильный ответ). Только при выполнении всех трех условий происходит переход системы к конечному состоянию, а именно – к сохранению вопроса и вариантов ответа к нему в базе данных. Начальное состояние системы изображается черным кругом, действия – прямоугольником со скругленными углами, условие – ромбом, конечное состояние системы – черным кругом в обводке.

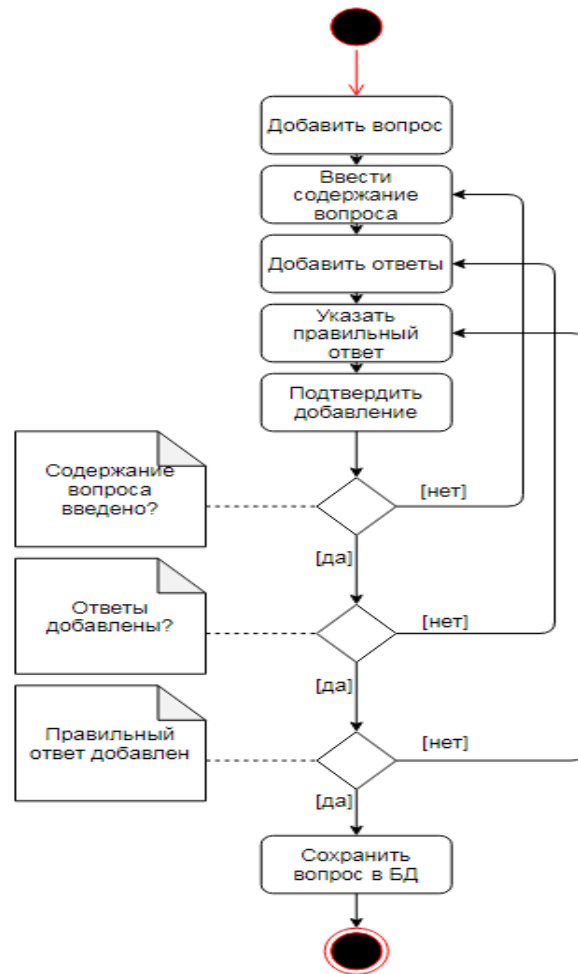


Рис. 2. Диаграмма деятельности прецедента «Добавить вопрос»

Для изображения взаимодействия между выделенными классами на этапе построения модели анализа разберем диаграмму последовательностей для прецедента «Добавить вопрос» (рис. 3).

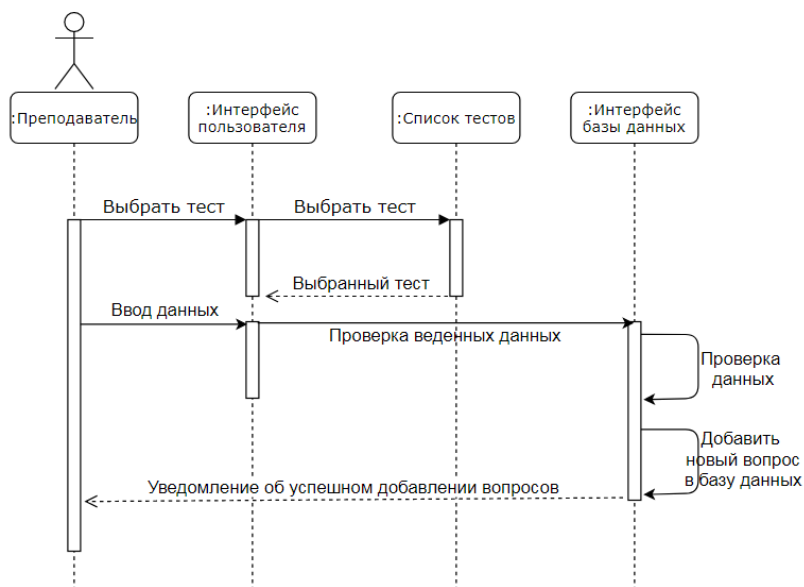


Рис. 3. Диаграмма последовательности прецедента «Добавить вопрос»

Диаграмма последовательности помогает установить временную взаимосвязь между классами, иными словами, она лишь показывает реализацию поведения прецедента, но не является точным представлением каждого его шага.

UML-диаграммы описывают концептуальную модель системы для создания тестов.

Обозначим функциональные требования к разрабатываемой системе:

1. При работе с системой преподаватель должен иметь возможность осуществлять следующие функции:

- создавать, удалять, редактировать темы;
- создавать, удалять, редактировать вопросы;
- создавать, удалять конфигурацию теста, которая включает в себя: номер студента, наименование теста, и количество вопросов;
- просматривать результаты тестирования по определенному тесту, а именно общую информацию, которая включает в себя: фамилию, имя, отчество, оценку.

Также необходимо иметь возможность просматривать более подробную информацию о студенте, который прошел тест, а именно: перечень вопросов попавших в тестирование, правильно ответил или нет, а также какие варианты ответов выбрал обучающийся.

2. При работе с системой студент должен иметь возможность:

- доступа через Интернет к тестовой системе под своей учетной записью;
- отвечать на вопросы теста;
- просмотреть результат пройденного теста.

Для уточнения требований к разрабатываемой системе онлайн-мониторинга целесообразно провести анализ аналогов системы. Существует масса систем, предназначенных для автоматизации тестирования знаний обучающихся.

Проанализируем программы автоматизированного тестирования. Выберем три критерия анализа: возможность работы через интернет; платформу и вид лицензии.

Система «СИНТеЗ»:

- имеет возможность работы через сеть Интернет;
- платформа: PHP, MySQL;
- лицензия данной программы распространяется на платной основе.

Есть возможность создавать вопросы различных типов, разбивать их на темы, в текст вопроса и ответа вставлять картинки, звук, фильмы, списки, таблицы и т.д.

Система «Конструктор тестов»:

- не имеет возможности работы через сеть Интернет;
- платформа: C++;
- лицензия данной программы распространяется на бесплатной основе.

В тестах существует возможность использовать музыкальное и звуковое сопровождение, графические изображения, видеоматериалы. Тестирование студентов проводится на одном компьютере поочередно. Каждый тестируемый регистрируется в программе под своим именем.

Система «OpenTest 2.0»:

- имеет возможность работы через сеть Интернет;
- платформа: HTML, PHP, JavaScript;
- лицензия данной программы распространяется на платной основе.

При проектировании модулей системы основное внимание было уделено высокой эргономичности системы, а также ориентации на работу с большим потоком пользователей.

В таблице 4 приведена сравнительная характеристика систем автоматизированного тестирования.

Таблица 4

Анализ аналогов

№	Название системы	Работа через интернет	Платформа	Лицензия
1	Система «СИНТеЗ»	+	PHP, MySQL	Платная
2	«Конструктор тестов»	-	C++	Бесплатная
3	«OpenTest 2.0»	+	HTML, PHP, JavaScript	Платная

В ходе анализа программ аналогов были выявлены недостатки у вышеперечисленных систем тестирования, а именно функциональная перегруженность, платная лицензия, а также не все системы являются масштабируемыми и не работают через интернет. Таким образом можем сделать вывод, что разработка собственной системы для создания тестов, лишенной выделенных у программ-аналогов недостатков является целесообразной.

ЛИТЕРАТУРА

1. Забродин, А. В. Основы проектирования информационных систем с помощью языка UML : учебное пособие / А. В. Забродин, В. П. Бубнов. – Санкт-Петербург : ПГУПС, 2018. –146 с.

2. Сибилева, И. А. Контроль знаний студентов как составляющая системы мониторинга качества образования / И. А. Сибилева // Вестник Мичуринского государственного аграрного университета. – 2011. – № 2-2. — С. 237-239.

3. Яковлев, А. Л. Современные методы и средства создания единого информационного пространства ВУЗа на примере ГОУ ВПО МГТУ / А. Л. Яковлев // Новые технологии. – 2009. – № 1. – С. 107-109.