**Министерство науки и высшего образования Российской Федерации**

**Университет ИТМО**

**Факультет инфокоммуникационных технологий**

Лабораторная работа №3

на тему

«Формализация требований»

по дисциплине:

«Проектирование инфокоммуникационных систем»

Выполнил:

Матвеев Геннадий Андреевич

группа K4110с

Санкт-Петербург

2020

Цель работы:

Для выбранного варианта инфокоммуникационной системы реализовать программные документы на создание системы, на основании которых будет организована реализация проекта.

**Задание 1. Создание документа-концепции.**

**Дополнительная спецификация**

**Даты внесения изменений**

|  |  |  |  |
| --- | --- | --- | --- |
| Версия | Дата | Описание | Автор |
| Черновой начальный вариант | 14.09.2020 | Черновой вариант, который можно будет уточнить на стадии развития | Матвеев Геннадий |

**Введение**

В этом документе описаны требования к системе тестирования студентов, не входящие в описание прецедентов.

**Регистрация событий и обработка ошибок**

Все ошибки регистрируются на сервере. Ведется журнал событий, который также хранится на сервере.

**Безопасность**

Необходимо выполнять аутентификацию всех пользователей.

**Человеческие факторы**

Система будет использоваться для тестирования не только продвинутых пользователей компьютера, но и также для людей, которые возможно не сильно «ладят» с информационными системами, так что процесс авторизации и само выполнение теста должны быть понятны, интуитивны, и иметь некие подсказки для пользователей.

**Надёжность**

Система должна быть стабильна, что подразумевает отсутствие проблем с авторизацией (не должно быть вылетов из учётной записи пользователя), надёжное сохранение всех действий и результатов пользователей за тест на сервере.

**Возможность восстановления информации**

При сбоях в работе, не зависящих от системы (проблемы с интернетом у пользователя, выключение устройства пользователя и т.д), а также при проблемах на сервере, система должна подстраиваться под проблемы. Например, если студент во время выполнения теста отключился по каким-либо причинам, система должна сохранить его промежуточный результат. А далее уже дать возможность преподавателю решить данный вопрос: дать студенту продолжить прохождение теста или начать выполнение заново, используя те же вопросы, или заново сгенерировав задания из банка заданий.

При возникновении проблем со стороны сервера необходимо продумать возможность сохранения результатов теста на устройстве студента. Другим вариантом развития событий при проблемах на сервере может быть подключение второго сервера. Однако данный вариант увеличивает сумму расходов на создание и поддержание системы. Его необходимо рассматривать исходя из задач, количества пользователей и объёма информации, находящейся в системе.

**Производительность**

Система тестирования студентов не обладает сложным и разнообразным функционалом, так что необходимо спроектировать и настроить её таким образом, чтобы она работала со скоростью, привычной для пользователей.

**Ограничения**

Руководство проекта настаивает на применении технологии Python, поскольку это весьма распространённый язык, часто применяемый в веб-сервисах. Использование данного языка даст возможности для дальнейшего усовершенствования проекта, а также обеспечит простоту разработки системы тестирования студентов.

**Приобретаемые компоненты**

Использование уже частично готовых средств для создания системы.

**Интерфейсы**

Монитор

Клавиатура

Мышь

Сенсорный экран

**Словарь терминов**

|  |  |  |
| --- | --- | --- |
| Термин | Определение | Синоним |
| Пользователь | Студент или преподаватель, использующий систему |  |
| Тестирование | Процесс проверки знаний студента путём сбора его ответов на вопросы |  |
| Функционал системы | Возможные для использования функции системы |  |

**Видение**

**Введение**

Нам видится надежная система тестирования студентов, обеспечивающая использование её с помощью разных устройств (телефон, компьютер, планшет, и т.д) на разных операционных системах (Windows, Mac OS, и т.д.), которая будет бесперебойно выполнять свои функции, иметь простой и понятный интерфейс, а также подготовленная для дальнейшего развития функционала.

**Формулировка проблемы**

Большинство современных систем тестирования студентов платны, бесплатные системы же как правило нужно точно также настраивать, пользуясь решениями сообщества данной системы. Нет чёткого понимая работы данных систем, имеются некоторые скрытые баги. Данная система тестирования будет создана индивидуально под нужды заказчика.

**Перспектива продукта**

Система будет использоваться для обучения будущих поколений студентов.

**Задачи уровня пользователя**

Преподаватель. Создаёт тесты, выбирая вопросы различных типов, задаёт параметры отчётов, выдаваемых студентам, исправляет уже созданные тесты, просматривает журналы оценок, а также контролирует ход сдачи тестов студентами.

Студент. Выполняет тесты, просматривает отчёты по тестам, просматривает прогресс по тестам.

Администратор. Выдаёт права с расширенными возможностями преподавателям, мониторит состояние системы.

**Преимущества системы**

|  |  |
| --- | --- |
| Свойство | Преимущества заинтересованных лиц |
| Система будет автоматически проверять ответы студентов | Преподавателям более не придётся проверять множество письменных работ студентов |
| Система будет автоматически выдавать студентам отчёты по их тестам | Преподавателям не придётся тратить время на информирование студентов об их оценках |
| Система будет хранить все данные о студентах и их успехах в прохождении тестов | Преподавателям не придётся самим составлять журнал оценок |
| Система будет выдать индивидуальные задания для каждого студента | Преподаватель будет знать, что студент работает самостоятельно |

**Основные свойства системы**

Авторизация пользователей

Управление пользователями администратором

Разграничение уровней доступа

Возможность создания вариативных тестов

Просмотр отчётов студентами

Настройка свойств отчёта студента преподавателем

Просмотр подробного журнала оценок студентов преподавателем

**Задание 2. Разработка спецификации требований к программному обеспечению (Modern Software Requirements Specification).**

**SRS**

**История изменений**

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Версия | Описание | Автор |
| 14.09.2020 | 1.0 | Исходная версия | Матвеев Геннадий |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Введение**

**Цель**

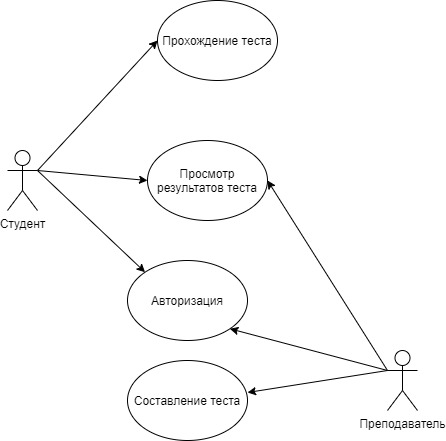
Создание системы, упрощающей работу преподавателя по проверке знаний у студентов. Обеспечение наглядной системы с оценками и прогрессом в прохождении тестов.

**Масштаб**

Разрабатывается в виде web-системы, используемой на различных устройствах в разных браузерах

**Краткая характеристика модели прецедентов**

|  |  |  |
| --- | --- | --- |
| **Название** | **Описание** | **Перечень акторов** |
| Прохождение теста | Студент выбирает правильный по его мнению ответ на тестовый вопрос или вбивает в поле ответ на данную в задании задачу | Студент, сервер, система |
| Просмотр результатов теста студентом | Студент просматривает результаты выполнения своих заданий (полнота отображения результатов зависит от настроек полноты отображения их для студента, заданных преподавателем) | Студент, сервер, система |
| Просмотр результатов теста преподавателем | Преподаватель просматривает всю сохраняемую информацию о тесте студента (оценка, баллы за тест, баллы за каждое задание, ответы студента на каждый вопрос, время затрачиваемое на каждый отдельный вопрос и на тест в целом) | Преподаватель, сервер, система |
| Авторизация | Пользователь (студент/преподаватель) вводит свои логин и пароль для входа в свою учётную запись | Пользователь (студент/преподаватель), сервер, система |
| Составление тестов | Загрузка преподавателем вопросов в банк заданий (хранящийся на сервере), заполнение условий тестов (количество вопросов какой сложности должно быть в одном тесте, время на тест) и условий отчётов для студентов (полнота отчёта) | Преподаватель, сервер, система |



**Характеристика акторов**

|  |  |
| --- | --- |
| Имя | Описание |
| Студент | Человек, использующий систему для выполнения тестов |
| Преподаватель | Человек, использующий систему для создания тестов и контроля студентов |
| Сервер | Серверное ПО, которое взаимодействует с системой и хранит её данные |
| Система | Средство, обеспечивающие возможность работать пользователем в системе |
| Администратор | Человек, следящий за работоспособностью системы |

**Требования**

**Функциональные требования**

• формировать тестовые задания индивидуально каждому студенту из банка заданий;

• доносить информацию до студентов о полученных, в ходе тестирования результатов;

• формировать отчетные материалы (списки, таблицы, графики) о результатах прохождения обучающего курса по каждому студенту по отдельности и о всех учебной группе в целом.

• содержать результаты о прохождении практических работ;

• хранить списки обучающихся студентов и уровня их успеваемости;

• хранить списки лиц с повышенным уровнем доступа (преподаватели, администраторы);

**Нефункциональные требования**

• удобный и простой пользовательских интерфейс;

• возможность организации двух разных режимов работы преподавателя и студента;

• используемое программное обеспечение должно быть постоянно обновляющимся, быть надежным и иметь возможность интегрирования новых функций;

• система должна поддерживать большое количество онлайн пользователей для корректной работы с тестированиями;

• обеспечивать пользователей сохранением их персональных данных при помощи системы паролей;

• поддержка работы в разных операционных системах.

**Ограничения проектирования**

В качестве языка программирования должен использоваться Python. Система должна работать стабильно и соответствовать современным стандартам по скорости работы.

**Закупаемые компоненты**

Возможна покупка сервера и всего необходимого для его подключения. (Другим вариантом является аренда сервера).

**Интерфейсы пользователя**

Интерфейс пользователя должен быть однообразен и понятен. Интерфейс должен подстраиваться под разные разрешения экрана и правильно отображаться на разных устройствах и разных ОС.

**Применяемые стандарты**

Система должна быть совместима со следующими ОС: Windows (только поддерживаемые пользовательские версии – на данный момент это Windows 10), Mac OS, Linux.

Для просмотра информации в сети интернет подойдет любой современный браузер:

* Opera, версия: 68.0.3618.63
* Yandex, версия: 20.4.2.197;
* GoogleChrome, версия: 81.0.4044.138;
* Firefox, версия: 76.0.1;
* Safari, версия: 5.1.7.

**Замечания, касающиеся авторских прав**

При сбое со стороны разрабатываемой системы ответственность несет администратор, который настраивал данную систему.

Все программные и графические материалы, принадлежат только компании и не подлежат бесплатному использованию.

**Вывод:** в ходе выполнения данной лабораторной работы были реализованы такие программные документы как документ-концепция, а также была разработана спецификация требований к программному обеспечению.