

**Заключение.** В статье представлено описание объекта автоматизации информационной системы. Предложены способы снижения трудоемкости документоведа посредством внедрения информационной системы. Представлена оптимизация основных функций. Показаны функциональные и концептуальная модели при проектировании будущей информационной системы.

## **СПИСОК ЛИТЕРАТУРЫ**

1. Разумников С. В. Модели, алгоритмы и программное обеспечение поддержки принятия стратегических решений к переходу на облачные технологии : монография / С. В. Разумников. — Томск : Издательство Томского политехнического университета, 2020. — 176 с. — Текст : непосредственный.
2. Разумников С. В. Некомпенсаторное агрегирование и рейтингование провайдеров облачных услуг / С. В. Разумников. — Текст : непосредственный // Доклады Томского государственного университета систем управления и радиоэлектроники. — 2018. — Т. 21, № 4. — С. 63-69.
3. Разумников С.В. Планирование развития облачной стратегии на основе применения многокритериальной оптимизации и метода STEM / С. В. Разумников. — Текст : непосредственный // Доклады Томского государственного университета систем управления и радиоэлектроники. — 2020. — Т. 23, № 1. — С. 53-61.

***Г. Б. БАРСКАЯ, А. А. КРИВОДАНОВА,  
С. О. СБРОДОВ, А. О. ШЕСТАКОВА***

*Тюменский государственный университет, г. Тюмень*

**УДК 004**

## **ИНФОРМАЦИОННАЯ СИСТЕМА РАСПОЗНАВАНИЯ И ПРОВЕРКИ ТЕСТОВЫХ ЗАДАНИЙ**

***Аннотация.** В работе представлены этапы проектирования и разработки информационной системы, позволяющей оптимизировать проверку тестовых заданий обучающихся и скорректировать тесты и учебный материал.*

*Рассмотрены способы снижения затрат времени и рисков допущения ошибок при проверке ответов обучающихся на тесты посредством внедрения автоматизированной информационной системы и анализа полученных данных.*

**Ключевые слова:** *информационная система, распознавание текста, анализ данных, формирование статистики, алгоритм проверки тестов.*

**Введение.** Первая официальная оценка знаний обучающихся при помощи тестов зафиксирована в Великобритании в 1864 г. Дж. Фишером. С каждым последующим годом динамика использования подобной оценки знаний возрастала, как и количество проверяемых работ. На настоящий момент тестирование обучающихся является повсеместной практикой оценки знаний и проводится на каждом предмете хотя бы раз в семестр/четверть. Преподаватели и учителя вынуждены вручную проверять каждую работу.

Существует система для распознавания текста SwiftScan — Scan PDF Documents [1]. Данная система позволяет сканировать печатный текст и преобразовывает его в цифровой формат.

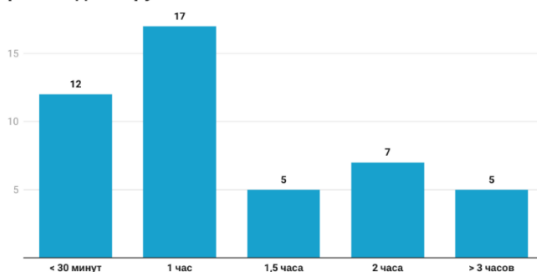
Для ведения успеваемости учеников используют электронные дневники. Одним из самых распространенных является цифровая образовательная платформа Дневник.ру [2], которая позволяет фиксировать оценки учеников, их данные и формировать необходимые отчеты.

Возможность создания и проверки тестов обеспечивают такие приложения, как Easy Test Maker [3]. Подобные системы позволяют проводить тестирование только в электронном формате.

**Проблема исследования.** В настоящее время существует множество систем для хранения результатов успеваемости обучающихся, но они не позволяют автоматизировать проверку ответов на тесты и сохранение результатов.

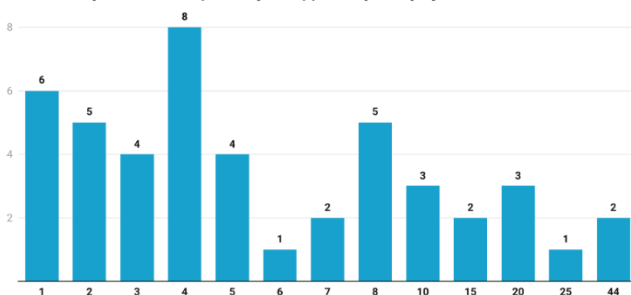
По данным проведенного опроса было выявлено, что средние затраты времени на проверку тестов одной группы занимает 1,3 часа (рис. 1), максимальное — 6 часов, среднее количество проверок в месяц — 6 (рис. 2). Таким образом, затраты времени преподавателя в месяц на проверку тестов составляют в среднем 7-8 часов на группу, в которой среднее количество человек — 25.

**Сколько примерно времени занимает у Вас проверка работ одной группы/класса?**



*Рис. 1.* Затраты времени на проверку работ одной группы

**Сколько раз в месяц Вы проводите проверку знаний?**



*Рис. 2.* Количество проверок знаний в месяц

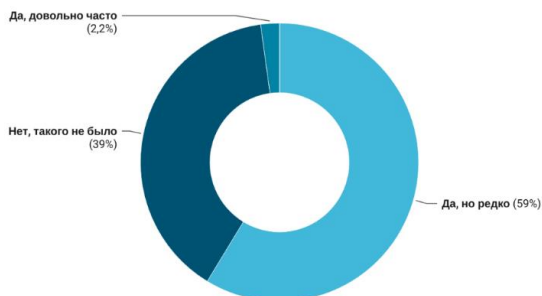
С ростом количества работ обучающихся, возрастает риск допущения ошибки при проверке ответов тестирования. По данным опроса было выявлено, что среднее количество проверяемых работ составляет от 100 до 250 (рис. 3) за одну тему тестирования и более 60% респондентов допускают ошибки при проверке работ обучающихся (рис. 4).

**Количество проверяемых работ**



*Рис. 3.* Количество проверяемых работ

**Бывало ли такое, что при проверке Вы ошибались в проверке правильности решения?**



*Рис. 4.* Процент преподавателей, допускающих ошибки при проверке

Целью исследовательской работы является разработка информационной системы, позволяющей сократить время преподавателя на проверку тестовых заданий и уменьшить риск возникновения ошибок при проверке. Для достижения поставленной цели следует выделить следующие задачи:

- составить требования, удовлетворяющие информационной системой;
- выбрать вариант решения проблемы;
- описать взаимодействие пользователя с системой;
- разработать структуру базы данных;
- разработать прототип информационной системы для решения проблемы.

**Материалы и методы.** Началом разработки информационной системы является формирование требований к реализации, которые были составлены на основе потребностей целевой аудитории:

- создание, редактирование и хранение списков групп, тестов и ответов;
- возможность загружать фотографии;
- возможность редактировать результат сканирования фотографии;
- формирование статистики ответов:

- по группе учащихся;
- по конкретному ученику;
- по результатам выполнения теста.

Сформирована концепция информационной системы, удовлетворяющая составленным требованиям: система позволяет загружать фотографии с рукописным ответом, преобразует в печатный вариант, осуществляет проверку тестовых заданий, позволяет просматривать статистику тестов с выявлением проблемных тем и вопросов.

Исходя из требований, система осуществляет возможность удаленного доступа к данным без необходимости использования строго определенного устройства.

Определены следующие критерии для выбора реализации:

- производительность;
- кроссплатформенность;
- работа с большими данными;
- скорость работы;
- возможность подключения нейронной сети.

Для создания визуального представления сайта используется HTML и CSS, в качестве фреймворка — Django на основе языка программирования Python [5, 6]. В качестве дополнительного языка JavaScript, необходимый для улучшения взаимодействия пользователя с сайтом и асинхронной передачи информации с базой данных.

Для хранения информации о пользователях, группах и тестах используется система управления базой данных SQLite [7]. Физическая модель базы данных представлена на рис. 5 [8].

Для сканирования рукописного текста фотографий с ответами выбран API Yandex.Cloud Vision [4], для использования которого необходимо выполнение следующих условий:

1. Файл изображения должен соответствовать требованиям:
  - поддерживаемые форматы файлов: JPEG, PNG, PDF;
  - максимальный размер файла: 1 МБ;
  - размер изображения не должен превышать 20 мегапикселей.

## 2. Кодировка файла Base64.

Для работы с API необходимо создать файл с телом запроса, например `body.json`. В свойстве `content` указано изображение, кодированное Base64.

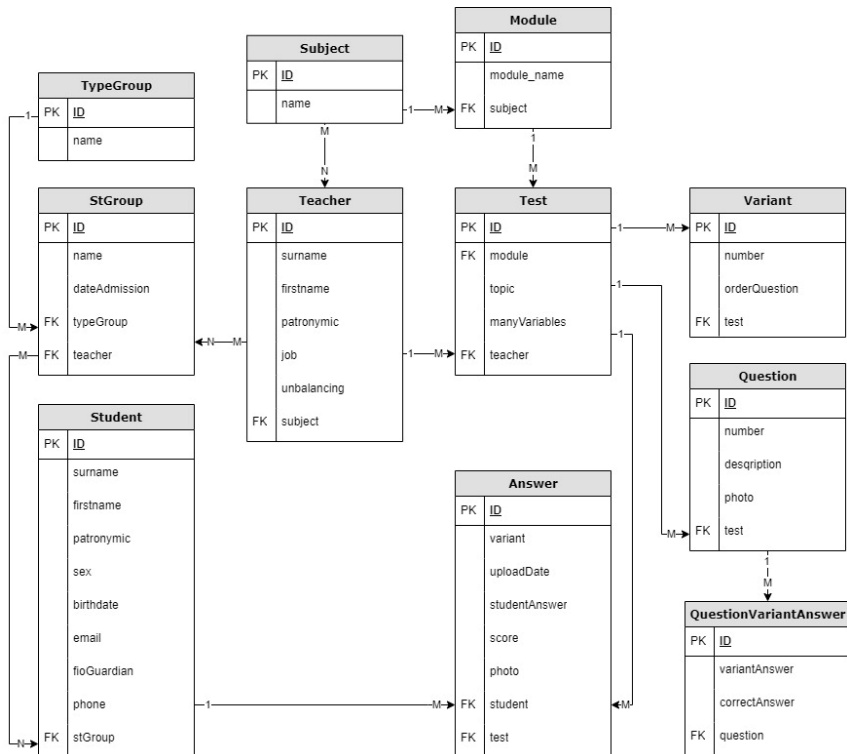


Рис. 5. Физическая модель базы данных

Файл конфигурации представлен на рис. 6.

Для распознавания фотографии с помощью метода `batchAnalyze` необходимо отправить запрос и сохранить ответ в файл. Пример кода запроса представлен на рис. 7.

```
body.json:
{
  "folderId": "b1gvmob95yysaplct532",
  "analyze_specs": [{
    "content": "iVBORw0KGgo...",
    "features": [{
      "type": "TEXT_DETECTION",
      "text_detection_config": {
        "language_codes": ["*"]
      }
    }
  ]
}]
}
```

*Рис. 6. Файл конфигурации*

```
$ export IAM_TOKEN=CggaATEVAgA...
$ curl -X POST \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer ${IAM_TOKEN}" \
  -d '@body.json' \
  https://vision.api.cloud.yandex.net/vision/v1/batchAnalyze > output.json
```

*Рис. 7. Код запроса к API*

Ответ будет состоять из распознанных блоков текста, строк и слов с указанием их местоположения на изображении.

Чтобы получить все распознанные слова с изображения, необходимо найти все строки со свойством "text". Например, с помощью использования функции grep: \$ grep -o "\"text\":\s\":[^\""]\*" output.json

Пример результата вывода:

```
"text": "PENGUINS", "text": "CROSSING", "text": "SLOW"
```

**Результаты.** Результатом исследовательской работы является приложение, удовлетворяющее требованиям к реализации.

Окно для загрузки фотографий с ответами показано на рис. 8. На рис. 9 представлено окно с результатами распознавания рукописного текста с возможностью редактирования ответов.

Данные для заполнения формы карточки теста формируются в json файл с помощью функции.

Введение в информатику | Устройство системы | Информатика

Модуль: Введение в информатику

Тема: Устройство системы

Предмет: Информатика

Выбор группы

дд.мм.гггг  Выберите файл | Файл не выбран | Сохранить ответы

Введение в информатику | Основы работы с компь | Информатика

Рис. 8. Форма загрузки фотографий

Распознано

Распознано Ученик		Вопрос	Ответ
Sbrodov	Sbrodov	1	B
Stanislav	Stanislav	2	A
		3	D

Рис. 9. Пример сканирования текста



Ответом стороне клиента является json файл с необходимыми данными для заполнения. Пример ответа представлен ниже:

```
json {'idTest': 2,
      'module': 'Введение в информатику',
      'topic': 'Устройство системы',
      'subject': 'Информатика',
      'questions': [[1, 'Файловая система — это:', ",
                    {'correct': [[4, 'вложенная структура папок и файлов']], 'wrong':
[[1, 'системная программа'], [2, 'вложенная структура файлов'], [3,
'вложенная структура папок']]}],
                    [2, 'В основные функции информатики как науки и технологии
входит:', ", {'correct': [[1, 'разработка методов исследования инфор-
мационных процессов']], 'wrong': [[2, 'построение технологий при-
менения (актуализации) знаний'], [3, 'только создание алгорит-
мов']]}]]]}
```

На рис. 10 представлена страница статистики, которая позволяет просматривать как статистику группы и отдельного ученика, так и общую статистику по тесту.

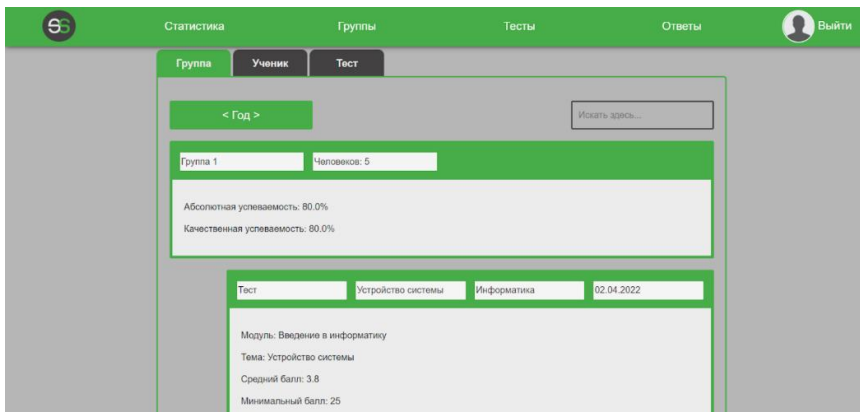


Рис. 10. Страница просмотра статистики

Данные для страницы статистики проходят через ряд вспомогательных функций, необходимых для удобного анализа результатов обучающихся.

Созданы страницы личного кабинета пользователя, страница для ведения групп аналогичная странице тестов.

В личном кабинете преподаватель указывает процент правильных ответов на формирование оценки. Например, от 61% — оценка «3», от 76% — оценка «4», от 91% — оценка «5», ниже 61% — оценка «2».

**Заключение.** Проанализировав потребительский рынок, выявлено, что на данный момент нет общедоступного решения, которое бы удовлетворяло всем требованиям.

В ходе выполнения исследовательской работы разработан прототип информационной системы распознавания и проверки тестовых заданий. Система позволяет сократить время преподавателя на проверку тестов и уменьшить риск возникновения ошибок при длительной оценке тестовых заданий.

Для разработки системы использован фреймворк Django на языке программирования Python и API Yandex.Cloud Vision. Для работы с данными — встроенная во фреймворк СУБД SQLite.

## СПИСОК ЛИТЕРАТУРЫ

1. SwiftScan: Scan PDF Documents : [сайт]. — URL: <https://swiftscan.app/en/index.html> (дата обращения: 30.03.2022). — Текст : электронный.
2. Дневник.ру Цифровая образовательная платформа : [сайт]. — URL: <https://dnevnik.ru/> (дата обращения: 30.03.2022). — Текст : электронный.
3. Easy Test Maker : [сайт]. — URL : <https://www.easytestmaker.com/html> (дата обращения: 30.03.2022). — Текст : электронный.
4. Yandex.Cloud Vision : [сайт]. — URL : <https://cloud.yandex.ru/docs/vision/> (дата обращения: 30.03.2022). — Текст : электронный.
5. Документация Django : [сайт]. — URL: <https://djbook.ru/rel3.0/> (дата обращения: 30.03.2022). — Текст : электронный.
6. Головатый А. Django. Подробное руководство / А. Головатый, Д. Каплан-Мосс. — Москва : Символ-плюс, 2014. — 512 с. — Текст : непосредственный.
7. Форсье Дж. Django. Разработка веб-приложений на Python / Дж. Форсье, П. Биссекс, У. Чан. — Москва : Символ-плюс, 2014. — 343 с. — Текст : непосредственный.
8. Мюллер Р. Дж. Базы данных и UML. Проектирование / Р. Дж. Мюллер. — Москва : ЛЮРИ, 2002. — 420 с. — Текст : непосредственный.