

М. В. Ступина, канд. пед. наук, доц., masamvs@bk.ru,
Донской государственный технический университет, Ростов-на-Дону

Архитектура программной системы поддержки учета посещаемости в вузе

В соответствии с модульно-рейтинговой системой, принятой в вузах, а также в силу смещения акцентов в сторону самостоятельной работы обучающихся и построения индивидуальных образовательных траекторий, актуализируется задача учета посещаемости. В статье рассмотрен проект системы, основной целью которой является автоматизация процесса подтверждения присутствия обучающихся на учебном занятии за счет сканирования QR-кода. Представлены результаты визуального моделирования, структура системы, организация хранения данных, а также алгоритмы работы основных программных модулей. Приведены типовые примеры работы веб-приложения и мобильного приложения как основных компонентов программной системы учета посещаемости.

Ключевые слова: учет посещаемости, электронный журнал, QR-код, диаграмма прецедентов, архитектура системы, мобильное приложение, веб-приложение, сервер обработки, СУБД MySQL, Swift, PHP

Введение

Стремительное развитие ИТ-сферы и средств информационно-коммуникационных технологий в последние годы привели к автоматизации множества областей человеческой деятельности, в том числе и системы высшего образования. Необходимость обеспечения конкурентоспособности университета и его соответствия мировым стандартам в области просвещения в современных условиях требуют цифровизации всей инфраструктуры вуза и реинжиниринга следующих его процессов в области автоматизации: административно-хозяйственной деятельности, управленческого учета, управления учебным процессом и т. д. [1, 2]. С этой целью вузы переходят на комплексные системы управления, позволяющие автоматизировать различные информационные процессы. К таким процессам можно отнести хранение, обработку и анализ информации в рамках деятельности вуза в различных областях, таких как поступление, обучение и оплата за обучение; расчет и распределение нагрузки; управление научной работой и инновациями; выполнение основных операций в личных кабинетах пользователей; деятельность на рабочем месте структурных подразделений (приемная комиссия, деканаты, отдел магистратуры и аспирантуры, диссертационные советы, управление дополнительным образованием и т. д.). При этом

все возрастающие требования к вузам со стороны государства требуют постоянной модернизации программно-аппаратного обеспечения, дополнения его новыми функциональными возможностями, а также создания специализированных интернет-сервисов и мобильных приложений.

Действующими Федеральными государственными образовательными стандартами, в соответствии с которыми осуществляется подготовка в рамках образовательных программ, определяются необходимость наличия в вузе системы оценки качества подготовки обучающихся. В связи с этим в вузах широкое применение получила модульно-рейтинговая система как форма контроля, которая, в частности, учитывает и посещаемость занятий обучающимися. Сказанное выше определяет необходимость решения задач хранения, предоставления, учета, анализа и т. д. информации о посещаемости обучающихся в электронном виде.

В то же время, как показывает практика, подтверждение присутствия обучающихся на занятии осуществляется путем проведения преподавателем переключки с последующим занесением информации в электронный журнал. Этот процесс является довольно рутинным, может занимать продолжительное время в случае большой аудитории слушателей, а также может сопровождаться ошибками, связанными с человеческим фактором. Например,

обучающийся может не услышать свою фамилию, преподаватель может пропустить обучающегося в списке группы или внести некорректные данные и т. п.), что актуализирует процесс поиска программных решений автоматизации процесса учета посещаемости в вузе.

Проблема исследования

Современные программные решения, существующие на рынке, предлагают ряд комплексных систем для вузов класса CRM (*Customer Relationship Management*), отвечающих за управление взаимоотношениями с клиентами и автоматизацию бизнес-процессов. В качестве примеров подобных CRM можно назвать ListOkCRM, Отмечалка, Таланто, BasePlan и др. Возможности CRM позволяют формировать журналы посещаемости, распределять учебную нагрузку, автоматически подготавливать необходимые статистические отчеты и т. д. Однако в CRM не в полной мере реализованы программные модули, непосредственно автоматизирующие процесс подтверждения присутствия обучающихся на учебном занятии. Кроме того, переход на CRM требует значительных материальных вложений и трудовых затрат на изменения структуры программно-аппаратных комплексов вуза.

На настоящее время все большее распространение получают технологии управления мобильными устройствами для создания мобильной образовательной среды в вузе. Так, например, в работе [3] представлены возможности использования мобильных устройств в целях обеспечения контроля посещаемости в автоматическом режиме на основании фактического местоположения обучающегося.

Особенности применения беспроводных технологий, таких как Bluetooth, Wi-fi, NFC и др. нашли отражений в работах [4–7]. Процесс внедрения комплекса мобильных средств поддержки учебного процесса (мобильные приложения "Расписание", "Распределение", "Успеваемость", "Посещаемость" и др.) рассмотрен в работе исследователей на базе Санкт-Петербургского политехнического университета [4]. Так, разработанное мобильное приложение "Посещаемость" позволяет автоматически регистрировать присутствие обучающегося на занятии на основании беспроводных технологий. Как отмечают авторы, это требует наличия в аудиториях специализированного оборудования.

В настоящее время автоматический учет посещаемости может быть реализован на базе алгоритмов детектирования и распознавания лиц.

В работе [8] представлено решение задачи детектирования с использованием алгоритма Виолы—Джонса и перспективы данного подхода в настройке эталонов студентов, учете посещаемости и его анализа. В исследовании [9] авторами подробно рассмотрены различные алгоритмы, методы и подходы к распознаванию лиц, представлена архитектура системы учета посещаемости и реализация системы с использованием искусственных нейронных сетей.

В качестве еще одного из наиболее перспективных направлений в области автоматизации процесса учета посещаемости учебных занятий можно выделить использование QR-кодов, получивших распространение во время пандемии коронавирусной инфекции COVID-19. Так, в работе [10] рассмотрены вопросы генерации QR-кода в форме PNG с использованием библиотеки PHP QR Code, сканирования QR-кода мобильным устройством обучающимися с использованием API, занесения данных о присутствии обучающегося на занятии в электронный журнал. Отмеченные автором основные направления использования QR-кодов в учебном процессе свидетельствуют о перспективах данного подхода к автоматизации процесса учета посещаемости.

Постановка задачи исследования

Необходимость разработки программной системы учета посещаемости обусловлена потребностью в автоматизации процессов сбора, передачи и хранения данных о посещаемости обучающихся.

С учетом современных тенденций в области распространения QR-кодов, обеспечивающих кодирование любой информации и возможностей ее оперативного распознавания с использованием камер мобильных устройств, оптимальным видится создание программной системы, включающей в себя:

- мобильное приложение для обучающихся, позволяющее выполнить сканирование QR-кода;
- веб-приложение для ведения электронного журнала преподавателем.

Основные функциональные возможности программной системы включают:

- хранение данных о посещаемости обучающихся;
- предоставление данных о посещаемости в табличном виде всем субъектам учебного процесса;
- оперативный доступ преподавателя к веб-приложению с любого устройства;
- настройка электронного журнала посещаемости и импорт оперативной информации об учебных дисциплинах и обучающихся;

- генерация QR-кода учебного занятия в консоли преподавателя;
- фиксация присутствия обучающегося на учебном занятии путем сканирования QR-кода камерой мобильного устройства.

Для разработки предложенной системы необходимо решить перечисленные далее задачи.

1. Разработать архитектуру программной системы учета посещаемости.
2. Выполнить проектирование базы данных электронного журнала посещаемости.
3. Разработать веб-приложение, обеспечивающее доступ к информации о посещаемости и позволяющее сгенерировать QR-код учебного занятия.
4. Разработать мобильное приложение, позволяющее выполнять сканирование QR-кода.

Архитектура программной системы учета посещаемости

Программная система учета посещаемости представляет собой совокупность трех компонентов (рис. 1):

- веб-приложение (используется для генерации QR-кода и получения доступа к электронному журналу посещаемости);
- мобильное приложение (используется для выполнения сканирования QR-кода и передачи

данных в электронный журнал, необходима авторизация);

- сервер обработки (используется для получения запроса от мобильного приложения через API).

Основными информационными потоками в системе являются: данные для авторизации преподавателя; данные об учебных дисциплинах; данные об обучающихся; id учебной дисциплины; данные для авторизации обучающегося; аутентификационные данные обучающегося; данные о присутствии на занятии; данные о посещаемости.

Следует отметить, что с сервером обработки пользователи программной системы напрямую не контактируют. Данный компонент программной системы обеспечивает работу разработанного API. В разработанной программной системе API умеет обрабатывать три следующих ключевых запроса.

1. Запрос на авторизацию обучающегося в мобильном приложении. В этом запросе сервер обработки получает логин и пароль пользователя, а возвращает ответ в виде JSON-документа, содержащий статус авторизации обучающегося в программной системе.

2. Запрос на внесение отметки о присутствии обучающегося на учебном занятии. В этом запросе сервер обработки получает аутентификационные данные обучающегося и уникальный иден-

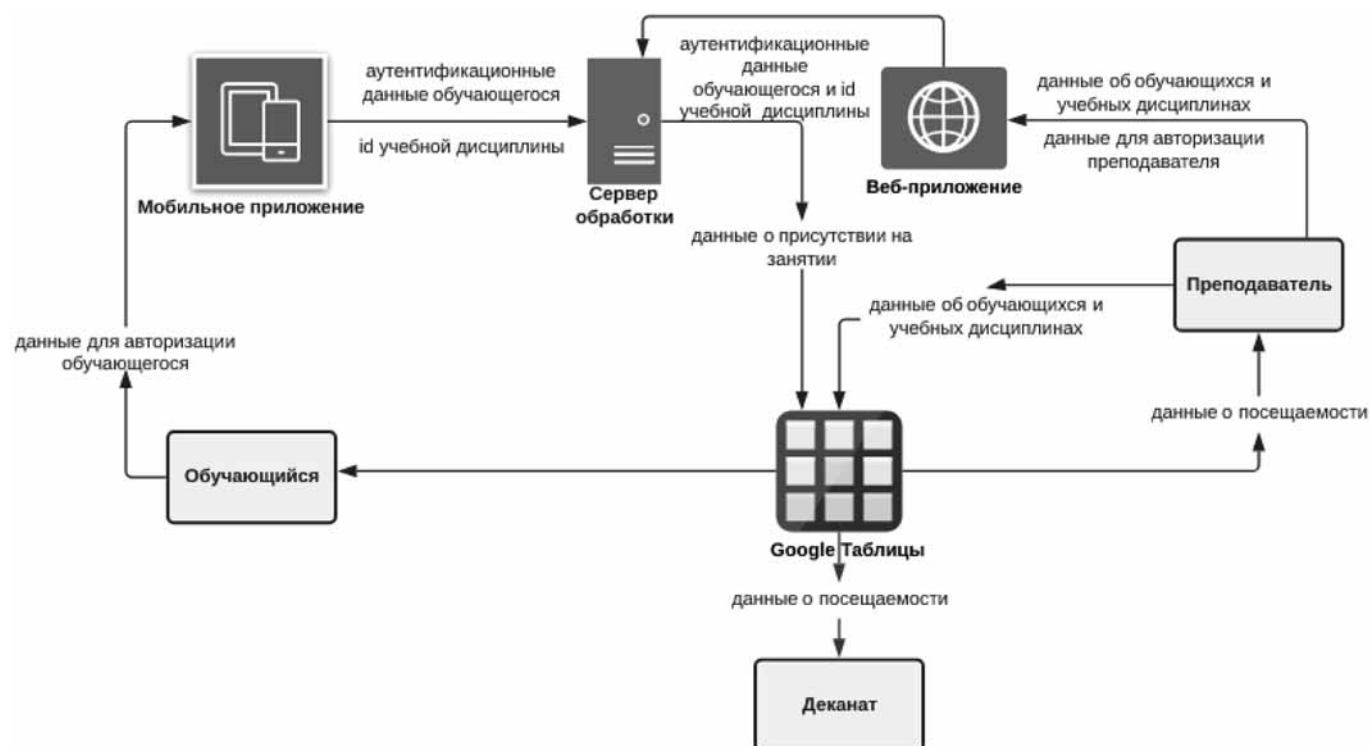


Рис. 1. Архитектура программной системы учета посещаемости

тификатор учебной дисциплины (соответствует spreadsheet id в Google Sheets API), полученный в результате сканирования QR-кода. Ответ сервера обработки данных содержит результат операции по внесению данных о присутствии.

3. Запрос на получение данных обучающегося. При данном запросе сервер обработки получает только аутентификационные данные обучающегося, а в качестве ответа возвращает массив данных (в виде JSON-документа), в котором указаны ФИО и учебная группа обучающегося.

Программная реализация системы

Моделирование системы. Проектирование программной системы выполнено на основе принципов объектно-ориентированного программирования с помощью языка моделирования UML. Для описания системы на концептуальном уровне была применена диаграмма вариантов использования (рис. 2).

Основные акторы: обучающийся, преподаватель, деканат. Варианты использования (прецеденты) представлены в таблице.

Хранение данных. Структура базы данных веб-приложения построена на основе реляционной модели данных (рис. 3).

Выделены четыре отношения:

1) Student_auth_data — авторизационные данные обучающихся:

- id — уникальный идентификатор;
- username — логин;
- password — пароль;

2) Student_personal_data — персональные данные об обучающихся:

- id — уникальный идентификатор;
- firstname — имя;
- lastname — фамилия;
- middlename — отчество;
- stud_group — учебная группа;

3) Teacher_auth_data — авторизационные данные преподавателей:

- id — уникальный идентификатор;
- username — логин;
- password — пароль;
- FIO — ФИО преподавателя;

4) Lessons — учебные дисциплины:

- id — уникальный идентификатор;

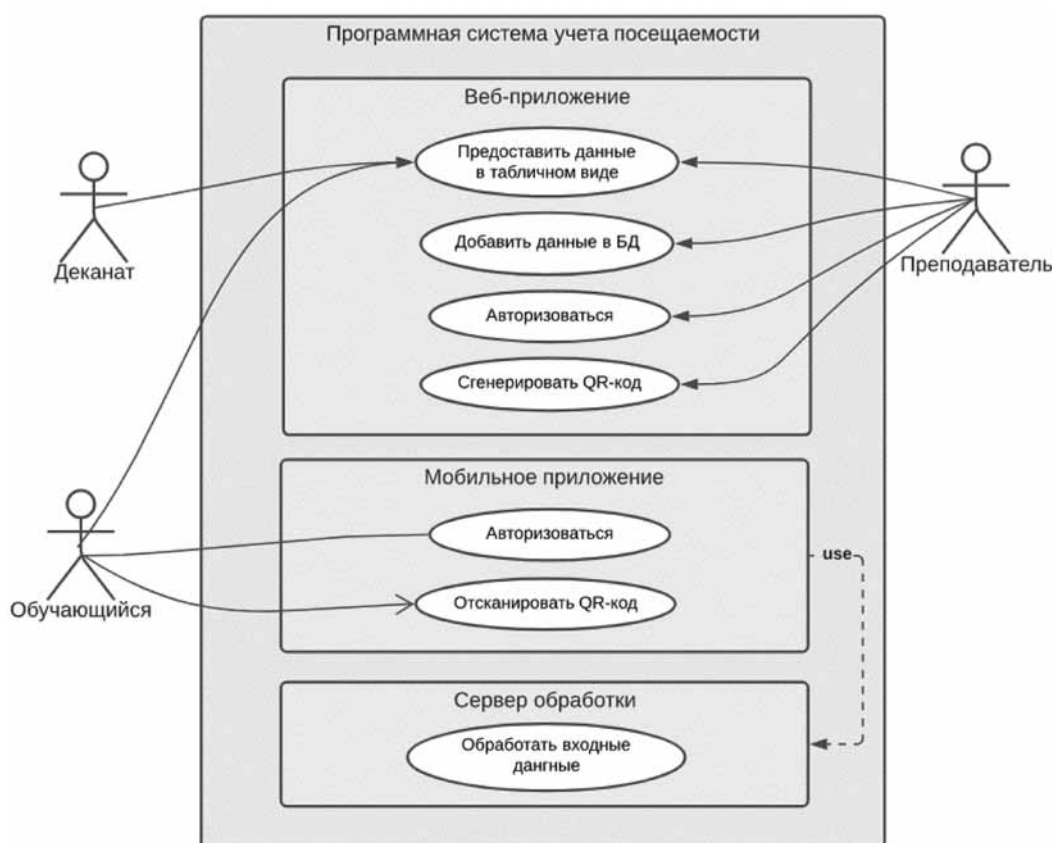


Рис. 2. Диаграмма вариантов использования

Описание прецедентов

| Вариант использования | Краткое описание |
|---|--|
| Предоставить данные о посещаемости в табличном виде | Запускается обучающимся, преподавателем, деканатом. Позволяет просмотреть информацию о посещаемости в электронном журнале |
| Авторизоваться в веб-приложении | Запускается преподавателем. Позволяет преподавателю авторизоваться в веб-приложении для дальнейшего взаимодействия с ним |
| Внести данные в базу данных | Запускается преподавателем. Позволяет преподавателю внести в базу данных данные об обучающихся, дисциплинах |
| Сгенерировать QR-код | Запускается преподавателем. Позволяет представить spreadsheetid учебной дисциплины в графическом виде (QR-код) |
| Авторизоваться в мобильном приложении | Запускается обучающимся. Позволяет обучающемуся авторизоваться в мобильном приложении для дальнейшего взаимодействия с ним |
| Отсканировать QR-код | Запускается обучающимся. Позволяет считать информацию с QR-кода и отправить ее на сервер обработки |
| Обработать входные данные | Используется мобильным приложением. Позволяет серверу обработки обработать переданные мобильным приложением данные |

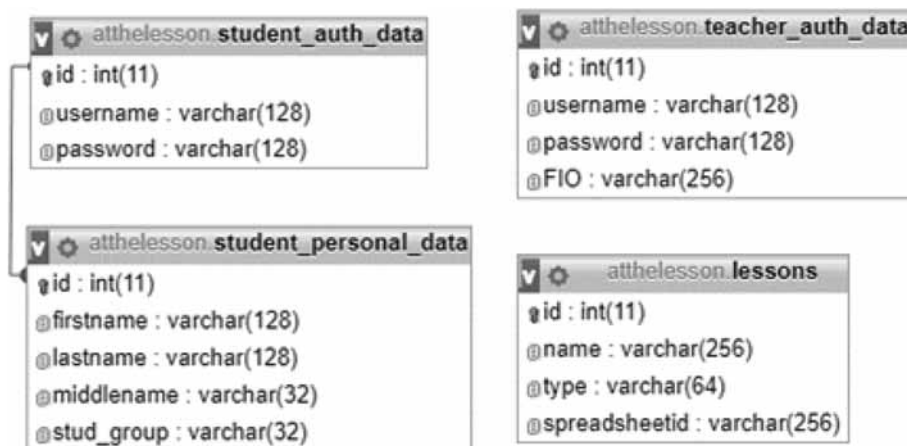


Рис. 3. Схема базы данных

- name — название дисциплины;
- type — тип учебного занятия;
- spreadsheetid — идентификатор Google Таблицы.

В качестве хранилища данных о посещаемости используются электронные таблицы Google Таблиц. Для учета посещаемости преподаватель создает несколько таких таблиц, а внутри таких таблиц — листы, в которые и заносятся данные о посещаемости. Каждый такой лист соответствует учебной группе, а каждая таблица — учебной дисциплине.

Модули веб-приложения. В качестве программных инструментальных средств разработки веб-приложения были выбраны: язык разметки HTML, язык описания стилей CSS, серверный язык программирования PHP и система управления базами данных MySQL. В качестве среды разработки использовался текстовый редактор Microsoft Visual

Studio Code с установленными расширениями для работы с HTML, CSS и PHP.

Электронный журнал, в котором отражена посещаемость обучающихся, представляет собой веб-приложение. Для работы с электронным журналом преподавателю необходимо авторизоваться, после чего станет доступна возможность просмотра и редактирования данных в специальной консоли. После настройки журнала, импорта данных о расписании занятий и учебных группах преподаватель имеет возможность сгенерировать QR-код, уникальный для каждого занятия.

На листинге 1 представлен фрагмент программного кода на языке PHP, отвечающий за генерацию QR-кода на соответствующей странице. В рассматриваемой программной системе используется генерация QR-кодов третьей версии, что означает создание QR-кода размером 29×29 модулей (точек).

```

if (isset($_REQUEST['data'])) {
    if (trim($_REQUEST['data']) == '')
        die ('Данные пустые, исправьте это!');
}
else {
    $filename = $PNG_TEMP_DIR.'test'.md5($_REQUEST['data'].'|'.
        $errorCorrectionLevel.'|'.$matrixPointSize).'.png';
    QRcode::png($_REQUEST['data'],$filename,
        $errorCorrectionLevel, $matrixPointSize, 2);
    echo '<hr/>';
    QRcode::png('Base Error Code Generation',$filename,
        $errorCorrectionLevel, $matrixPointSize, 2);
}

```

Листинг 1. Генерация QR-кода

Выбор размера при генерации отвечает за масштабирование генерируемого QR-кода в целях удобного вывода как на большие экраны, так и на экраны мобильных телефонов.

Модули мобильного приложения. Разработка мобильного приложения осуществлялась на языке Swift. В качестве среды разработки был выбран инструмент Xcode.

Обучающимся необходимо авторизоваться в мобильном приложении программной системы, после чего для них станет доступен функционал по сканированию QR-кода.

На листинге 2 представлен фрагмент программного кода обработки мобильным приложением ответа от сервера обработки на языке Swift.

Сканирование QR-кода, который был сгенерирован преподавателем, позволяет идентифицировать присутствие обучающегося на занятии. После сканирования QR-кода мобильное приложение выполняет запрос к серверу обработки данных через API. Ответ сервера обработки данных в JSON-формате будет содержать сообщение об успешном внесении данных о присутствии на занятии, либо сообщение об ошибке.

```

var qrData: QRData? = nil {
    didSet {
        if qrData != nil {
            MarkMe(onDone: { (response) in
                self.result = response.summary
                if (self.result! == "successfully"){
                    self.showDialogue(text: "Вы отметили
                    свое присутствие на паре!", title: "Успешно!")
                } else {
                    self.showDialogue(text: "Что-то пошло не так. Попробуйте еще раз.", title: "Ошибка!")
                },
                onError: { (error) in
                    self.showDialogue(text: "Отсканированный код
                    не является корректным.\nОбратитесь к
                    преподавателю.", title: "Ошибка!") },
                spreadsheetId: qrData!.codeString!,
                nickname: UserDefaults.standard.string(forKey: "nickname")!,
                password: UserDefaults.standard.string(forKey: "password")!
            })
        }
    }
}
?>

```

Листинг 2. Обработка ответа от сервера

Результаты исследования

На рис. 4 представлена главная страница веб-приложения, позволяющая преподавателю авторизоваться в веб-приложении программной системы учета посещаемости.

После авторизации становится доступной консоль преподавателя (рис. 5), где можно внести из-

менения в электронный журнал, выполнить импорт учебных дисциплин и списка обучающихся учебной группы, сгенерировать QR-код, а также сделать ряд других настроек.

На рис. 6 представлен фрагмент консоли, где преподаватель может добавлять, редактировать и удалять учебные дисциплины.

Консоль преподавателя также содержит страницу генерации QR-кода. Для того чтобы сгенерировать QR-код, который должны будут отсканировать обучающиеся, преподавателю нужно перейти на страницу генерации QR-кода, выбрать учебную дисциплину, вид занятия и размер генерируемого кода, а затем нажать на кнопку "Сгенерировать QR-код". Это действие преподаватель выполняет перед каждым учебным занятием.

Пример генерации для одной из лекций по дисциплине "Информационные технологии" QR-кода размером 10×10 мм — наименьшим размером, воспринимаемым камерой мобильного устройства, — представлен на рис. 7.

Обучающиеся проходят авторизацию в мобильном приложении. В случае неверного ввода данных будет открыто всплывающее окно, которое сообщает об ошибке. При этом аутентификационные данные обучающегося сохраняются, что при повторном запуске мобильного приложения позволяет сразу открыть страницу-сканер QR-кода.

Страница-сканер QR-кода является главным экраном мобильного приложения, позволяюще-

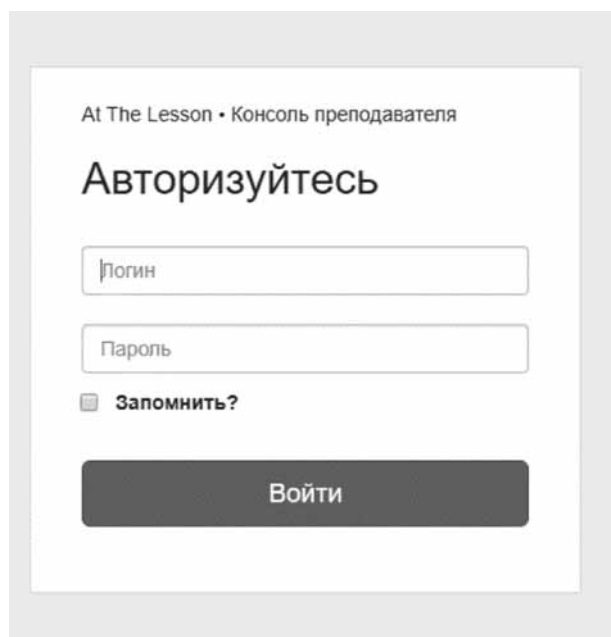


Рис. 4. Экран авторизации веб-приложения

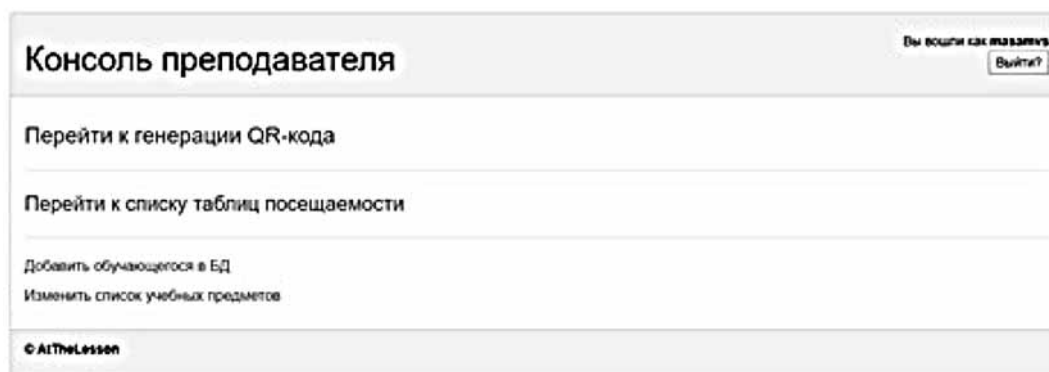


Рис. 5. Консоль преподавателя

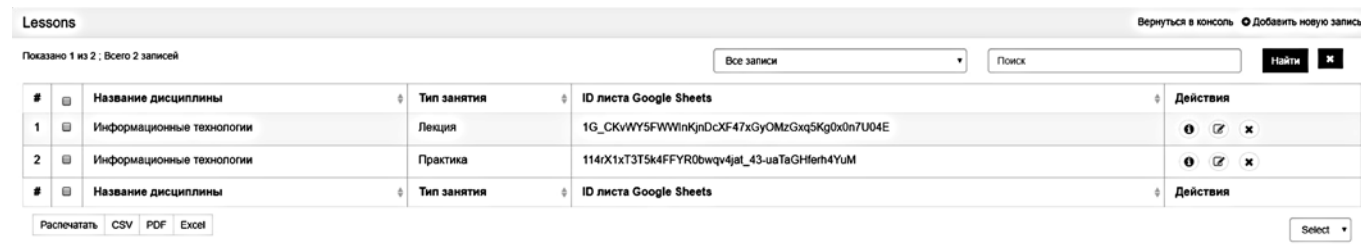


Рис. 6. Отображение списка учебных дисциплин в консоли преподавателя



Рис. 7. Сгенерированный QR-код

го отметить свое присутствие на занятии. Чтобы отсканировать QR-код, сгенерированный преподавателем и выведенный на экран проектора, обучающийся должен привести камеру мобильного устройства, на котором установлено мобильное приложение, на QR-код. Затем мобильное приложение автоматически отсканирует QR-код и запустит процесс отправки данных на сервер обработки. В случае, если обучающийся отсканирует недействительный QR-код, приложение выведет соответствующее всплывающее окно с ошибкой. Если же обучающийся отсканирует корректный QR-код, и он является обучающимся группы, для которой был сгенерирован QR-код в рамках учебной дисциплины, то мобильное

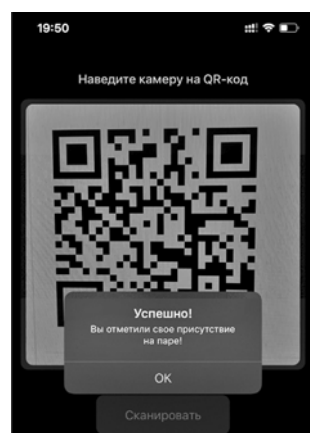


Рис. 8. Экран с успешной отметкой присутствия

приложение выведет соответствующее всплывающее окно, которое сообщит об успешной отметке о присутствии на учебном занятии (рис. 8).

В дальнейшем преподавателю доступна информация о посещаемости обучающихся (персональные данные скрыты в соответствии с Федеральным законом "О персональных данных" от 27.07.2006 № 152-ФЗ) на учебном занятии в электронном журнале (рис. 9).

Электронный журнал на базе облачного сервиса Google Таблицы позволяет формировать отчеты и статистику посещения учебных занятий, а сто-

| № | ФИО | 07.02 | 14.02 | 21.02 | 28.02 | 07.03 | 14.03 | 21.03 |
|----|-------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 1 | ##### | присутствует | присутствует | присутствует | присутствует | присутствует | | присутствует |
| 2 | ##### | | | | присутствует | присутствует | присутствует | присутствует |
| 3 | ##### | | | | | | | |
| 4 | ##### | присутствует | | присутствует | присутствует | присутствует | присутствует | |
| 5 | ##### | присутствует | | присутствует | присутствует | присутствует | присутствует | |
| 6 | ##### | присутствует | присутствует | присутствует | присутствует | присутствует | присутствует | присутствует |
| 7 | ##### | присутствует | присутствует | присутствует | присутствует | присутствует | присутствует | присутствует |
| 8 | ##### | | присутствует | присутствует | присутствует | присутствует | присутствует | присутствует |
| 9 | ##### | | присутствует | присутствует | присутствует | присутствует | | присутствует |
| 10 | ##### | | присутствует | присутствует | присутствует | присутствует | присутствует | присутствует |
| 11 | ##### | | | присутствует | присутствует | присутствует | присутствует | присутствует |
| 12 | ##### | | | присутствует | присутствует | присутствует | присутствует | |
| 13 | ##### | присутствует | | | | | | присутствует |
| 14 | ##### | | | | | | | присутствует |
| 15 | ##### | присутствует | присутствует | присутствует | присутствует | присутствует | | присутствует |
| 16 | ##### | присутствует | | | | присутствует | | присутствует |
| 17 | ##### | присутствует | присутствует | присутствует | присутствует | присутствует | | присутствует |
| 18 | ##### | | | | | присутствует | присутствует | присутствует |
| 19 | ##### | присутствует | присутствует | присутствует | присутствует | присутствует | | присутствует |
| 20 | ##### | присутствует | присутствует | присутствует | присутствует | присутствует | | присутствует |
| 21 | ##### | присутствует | присутствует | присутствует | присутствует | присутствует | | присутствует |
| 22 | ##### | | | | | | | |
| 23 | ##### | | присутствует | | | | | |
| 24 | ##### | присутствует | присутствует | присутствует | присутствует | присутствует | | |

Рис. 9. Фрагмент электронного журнала посещаемости

ронные дополнения и поддержка скриптов делают возможным расширение базового функционала в соответствии с потребностями субъектов образовательного процесса.

Заключение

Необходимость контроля освоения обучающимися образовательной программы, выполнение ими индивидуальных планов, в том числе и посещение учебных занятий, предусмотренных учебным планом, определяет необходимость со стороны преподавателя ведения учета посещений учебных занятий обучающимися.

Процесс учета посещаемости одной группы или потока при использовании программной системы учета посещаемости требует минимальных временных затрат, включая время на запуск веб-приложения и генерацию QR-кода, доступного для сканирования обучающимися.

Следует отметить, что несмотря на возможность передачи и распространения в электронном виде QR-кода учебного занятия среди студентов, фальсификация данных о присутствии на занятиях может быть проведена и при других способах учета посещаемости. Например, при проведении потоковой лекции в аудитории более 60 человек, переключка со стороны преподавателя может сопровождаться откликом одного студента на чужую фамилию — фамилию отсутствующего студента. Процесс учета посещаемости путем рукописной записи в списке присутствующих группы может быть сфальсифицирован за счет изменения почерка. Журналы посещаемости, заполняемые старостами, также могут быть исправлены в пользу отсутствующих студентов. Таким образом, при указанных способах учета посещаемости данные могут быть недостоверны, а их проверка с последующим заполнением электронного журнала требует дополнительного аудиторного времени или личного времени преподавателя. В то же время, использование QR-кодов, несмотря на возможность фальсификации данных о посещаемости со стороны

некоторых студентов, позволяет в наибольшей степени ускорить и автоматизировать процесс учета посещаемости.

Дальнейшее развитие программной системы учета посещаемости связано с разработкой кросс-платформенного мобильного приложения, а также интеграцией данного программного комплекса в единую информационно-образовательную среду вуза.

Список литературы

1. Новгородова Н. Г., Редькина Б. А. Автоматизация образовательной деятельности профессионально-педагогического вуза // Международный журнал экспериментального образования. 2014. № 3-2. С. 15—17.
2. Зафиевский А. В. Автоматизация управления учебным процессом в вузе // Успехи современного естествознания. 2010. № 1. С. 115—117.
3. Колосков С. С., Бабешко В. Н., Самочадин А. В., Кошлич Ю. А. Методика апробации внедрения мобильных сервисов на платформе управления мобильными устройствами в вузах // Экономика. Информатика. 2015. № 19 (216). С. 152—159.
4. Райчук Д. Ю., Самочадин А. В., Носницын С. М., Хмельков И. А. Комплекс мобильных средств поддержки учебного процесса // Информатика, телекоммуникации и управление. 2014. № 6 (210). С. 7—15.
5. Bhalla V., Singla T., Gahlot A., Gupta V. Bluetooth Based Attendance Management System // IJITET. 2013. Vol. 3, Iss. 1. P. 524—527.
6. More V., Nayak S. Attendance Automation using Near Field Communication (NFC) Technology // Internat. J. of Scientific & Engineering Research. 2013. Vol. 4, Iss. 12. P. 572—575.
7. Ayu M. A., Ahmad B. I. TouchIn: An NFC Supported Attendance System in a University Environment // Internat. J. of Information and Education Technology. 2014. Vol. 4, No. 5. P. 448—453.
8. Синецкий Р. М., Гавриков М. М. Система учета посещения занятий студентами на основе алгоритмов распознавания лиц // Известия вузов. Северо-Кавказский регион. Серия: Технические науки. 2016. № 3 (191). С. 24—30.
9. Иванова Е. В., Струева А. Ю. Система учета посещаемости студентов на основе распознавания лиц // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2021. Том 10, № 4. С. 60—73.
10. Stupina M. V., Anistratenko K. V., Pazina L. O. Using the QR code as a means of automating the process of accounting for attendance at educational classes // Journal of Physics: Conference Series. 2021. Vol. 2131. Article 022077. URL: <https://iopscience.iop.org/article/10.1088/1742-6596/2131/2/022077/pdf>

The Architecture of the Software System for Supporting Attendance Records at the University

M. V. Stupina, masamvs@bk.ru,

Don State Technical University, Rostov-on-Don, 344003, Russian Federation

Corresponding author:

Mariya V. Stupina, Associate Professor,

Don State Technical University, Rostov-on-Don, 344003, Russian Federation

E-mail: masamvs@bk.ru

Received on April 07, 2022

Accepted on May 11, 2022

Nowadays, in accordance with the modular rating system adopted in universities, as well as due to the shift in emphasis towards the independent work of students and the construction of individual educational trajectories, the task of accounting for attendance at classes is being updated. The purpose of this paper is to develop the architecture of a software system that allows you to automate the process of confirming the presence of students in the classroom. Based on the analysis of modern works and the technical capabilities of current software solutions, the method of scanning the unique QR code of each training session with the camera of the student's mobile device was chosen as the most optimal means of solving this problem. The paper presents the results of visual modeling of the software system, as well as the structure of the system, which includes the teacher's console in the form of a web application in PHP and a mobile application for the iOS platform developed in Swift. The organization of data storage in the MySQL DBMS, the structure of relations are described. The main algorithms for the operation of the main software modules for scanning and generating a QR code are considered. Test examples of the operation of a web application and a mobile application as the main components of the system are given. With a high degree of confidence, we can assume that the presented system will be effective for solving attendance accounting problems.

Keywords: attendance records, electronic journal, QR code, precedent diagram, system architecture, mobile application, web application, processing server, MySQL DBMS, Swift, PHP

For citation:

Stupina M. V. The Architecture of the Software System for Supporting Attendance Records at the University, *Programmная Ingengeria*, 2022, vol. 13, no. 7, pp. 344—353.

DOI: 10.17587/prin.13.344-353

References

1. Novgorodova N. G., Redkina B. A. Automation of educational activities of a vocational pedagogical university, *Mezhdunarodnyy zhurnal eksperimental'nogo obrazovaniya*, 2014, no. 3-2, pp. 15—17 (in Russian).
2. Zafievskij A. V. Automation of educational process management at the university, *Uspekhi sovremennogo estestvoznaniya*, 2010, no. 1, pp. 115—117.
3. Koloskov S. S., Babeshko V. N., Samochadin A. V., Koshlich Yu. A. Methodology for approbation of the implementation of mobile services on the mobile device management platform in universities, *Economy. Informatics*, 2015, no. 19 (216), pp. 152—159 (in Russian).
4. Rajchuk D. Yu., Samochadin A. V., Nosnicyn S. M., Hmel'kov I. A. A set of mobile means of supporting the educational process, *Informatika, telekommunikacii i upravlenie*, 2014, no. 6 (210), pp. 7—15 (in Russian).
5. Bhalla V., Singla T., Gahlot A., Gupta V. Bluetooth Based Attendance Management System, *IJIET*, 2013, vol. 3, iss. 1, pp. 524—527.
6. More V., Nayak S. Attendance Automation using Near Field Communication (NFC) Technology, *Internat. J. of Scientific & Engineering Research*, 2013, vol. 4, iss. 12, pp. 572—575.
7. Ayu M. A., Ahmad B. I. TouchIn: An NFC Supported Attendance System in a University Environment, *Internat. J. of Information and Education Technology*, 2014, vol. 4, no. 5, pp. 448—453.
8. Sineckij R. M., Gavrikov M. M. Student attendance system based on face recognition algorithms, *Izvestiya vuzov. Severo-Kavkazskij region. Seriya: Tekhnicheskie nauki*, 2016, no. 3 (191), pp. 24—30 (in Russian).
9. Ivanova E. V., Strueva A. Yu. Student attendance system based on face recognition, *Vestnik Yuurgu. Seriya: Vychislitel'naya matematika i informatika*, 2021, vol. 10, no. 4, pp. 60—73 (in Russian).
10. Stupina M. V., Anistratenko K. V., Pazina L. O. Using the QR code as a means of automating the process of accounting for attendance at educational classes, *Journal of Physics: Conference Series*, 2021, vol. 2131, article 022077, available at: <https://iopscience.iop.org/article/10.1088/1742-6596/2131/2/022077/pdf>