УДК 004.4

**ОБЗОР ПРИЛОЖЕНИЯ, СВЯЗАННОГО С АВТОМАТИЗАЦИЕЙ ДЕЯТЕЛЬНОСТИ IT-ОТДЕЛОВ ПРЕДПРИЯТИЯ**

К.В.Парахин – студент, Институт информационных технологий и электроники, кафедра ИСПИ, группа ПРИ-120, E-mail: parahinkv@gmail.com

С.В.Курочкин – научный руководитель, к.т.н., Институт информационных технологий и электроники, кафедра ИСПИ, E-mail: s-2000-k@yandex.ru

**Аннотация:** В статье рассматривается предметная область, связанная с деятельностью IT-отделов предприятия и взаимодействием сотрудников внутри ее. Исследуется актуальность разработки мобильного приложения, связанного с автоматизацией и оптимизацией данной деятельности, реализованного автором статьи в рамках выпускной-квалификационной работы при работе на предприятии ООО «АИСТСОФТ».

**Ключевые слова:** программно-информационная система, мобильное приложения, автоматизация деятельности, разработка приложения, IT-предприятие.

Каждое предприятие должно иметь хотя бы один IT-отдел, занимающийся разработкой, внедрением и поддержкой программно-информационных систем. Для автоматизации деятельности отдела необходимо иерархически разбить его на отдельные группы, каждая из которых имеет своего менеджера (руководителя), обладающего отдельными полномочиями по надзору и контролю подчиненных сотрудников. Каждый такой отдел при этом должен технически поддерживаться, чтобы вовремя находить и исправлять получающиеся неисправности. Поэтому можно прийти к выводу, что данное мобильное приложение действительно может стать очень удобным, полезным и применим при внедрении его в работу как нового, так и уже существующего IT-предприятия.

При его использовании проект компании может получить такие необходимые возможности, как:

1. Создание иерархической организационной структуры проектов внутри IT-отдела
2. Автоматизация деятельности и взаимодействия сотрудников IT-отдела
3. Введение автоматической системы отчетности и регистрации инцидентов
4. Хранение данных о сотрудниках, их мероприятиях, задачах, сообщениях
5. Оптимизация работы отделов мониторинга и аналитики

Разработка данного мобильного приложения действительно актуальна, следовательно, необходимо определить список функциональных и нефункциональных требований, предъявляемых к приложению:

Основные функциональные требования, предъявляемые к системе:

1. Использование авторизации пользователей на сервере;
2. Использование регистрации с подтверждением по почте;
3. Возможность планирования пользовательских мероприятий и приглашению гостей;
4. Возможность трекинга задач и уведомления связанных пользователей;
5. Возможность создавать автоматические и ручные отчеты о работе системы.

Основные нефункциональные требования:

1. Обеспечение уровня производительности – трафика запросов не менее 10 000 запросов к серверу в секунду;
2. Отказоустойчивость и надежность работы системы, регистрация неисправностей на сервере;
3. Работа на облачном Linux сервере, развернутом в контейнере;
4. Масштабируемость компонентов системы;
5. Платформо-ориентированность мобильного приложения (запуск на OC Android).

Для разработки серверной части приложения использовался фреймворк ASP.NET Core на платформе .NET 7. «Компания Microsoft старается по возможности обеспечивать обратную совместимость при выпуске новой версии программных продуктов»[3], поэтому большинство средств платформы .NET будут актуальны ближайшие годы.

Для разработки клиентской части приложения использовался фреймворк Flutter.

Для работы с хранилищем данных использовался ORM – фреймворк Entity Framework и СУБД Postgre SQL. «Postgre SQL представляет собой свободно распространяемую систему управления реляционными базами данных с открытым исходным кодом» [4], хорошо поддерживаемым на популярных языках программирования, например, на C#.

Развертывание и контейнеризация серверного приложения выполнялась с использованием с использованием технологии compose и программного обеспечения Docker. «Docker выступает как в качестве инструмента для создания и размещения контейнера в облаке, так и в качестве оркестратора» [5] – выполняющего обязанности проведения взаимодействия между контейнером сервера .NET приложения и контейнером PostgreSQL базы данных.

Диаграмма развертывания данной программной системы представлена ниже на рисунке 1 – на ней представлено взаимодействие Docker-контейнера серверного приложения и клиентского приложения, запущенного на Android-устройстве:

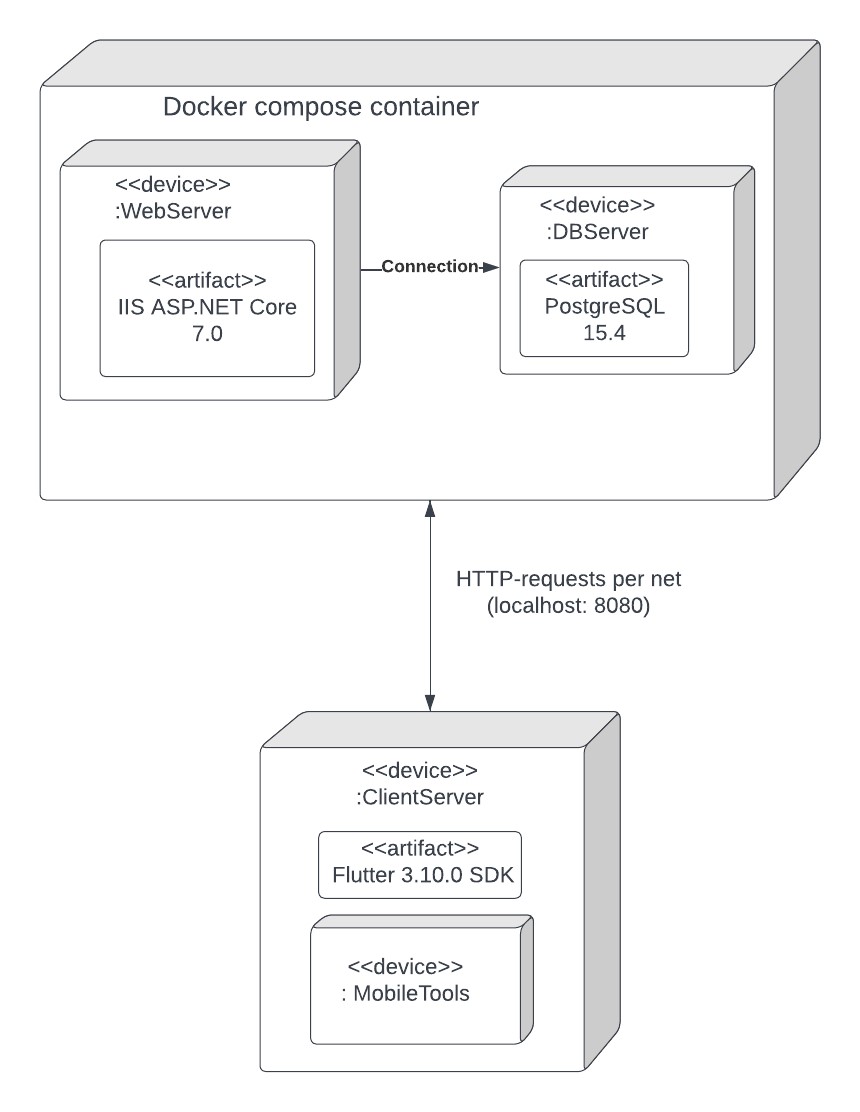


Рисунок 1 - Диаграмма развертывания программной системы

«Разработанное мобильное приложение представляет собой распределенное клиент-серверное приложение. Оно включает в себя облачное серверное приложение, которое имеет собственный жизненный цикл»[1]. Клиентские приложения при этом могут быть установлены пользователями на свои мобильные устройства Android – а далее осуществлять взаимодействие с сервером по протоколу REST.

Данные передаются между клиентским и серверным приложением с использованием HTTP-запросов в формате JSON. Часть из этих данных мобильное приложение сразу может отобразить для пользователя в визуальном виде, а часть – сохранить в локальной памяти устройства – и затем предоставлять их при посещении связанных страниц.

Благодаря этому приложение не делает слишком много запросов – но при этом и не перегружает сервер и его базу данных большим набором транзакций в рамках проведения единого большого запроса. Кэширование и наличие локальной памяти клиентского устройства дает возможность обеспечить требование производительности и адаптивности работы системы.

Также отдельно хотел бы рассмотреть логику мобильного приложения, связанной с рассылкой push-уведомлений пользователям. Данная возможность реализована благодаря использованию Firebase. «Во время авторизации в сессии пользователь выполняет запуск Firebase-приложения на своем Android-устройстве»[2]. На облачном Google-сервере генерируется уникальный Firebase-токен – по которому пользователя становиться возможным идентифицировать в системе. Затем, когда клиентское приложение отправляет результат авторизации – оно к тому же отправляет на сервер полученный токен. При этом стоит отметить, что в качестве мер безопасности - токены сохраняются в локальной памяти устройства – но в течении времени неактивности могут быть автоматически очищены.

Сервер приложения при этом работает в режиме 24/7 и осуществляет проверку изменений данных в базе – и если он обнаружит, что какие-то связанные с пользователем данные были обновлены – то он создает push-уведомление и отправляет его на все связанные с пользователем авторизованные firebase-устройства.

По итогам данного исследования можно сделать вывод о том, что текущий IT-рынок является достаточно конкурентоспособным, поэтому, чтобы вывести свой проект на новый уровень – нужно стремиться к увеличению качества и показателей качества работы за счет ведения мероприятий по автоматизации и оптимизации.

На рынке программных продуктов уже существует ряд программных систем (обычно выпускаемых на платной или подписочной основе), выпущенных как общие решения по автоматизации работы огромных информационных проектов. При этом для функционирования небольших IT-проектов обычно достаточно более компактной и проверенной системы - поэтому наиболее подходящим решением является самостоятельная разработка программной системы, которая выполняла вышеописанные функции.

**Список используемой литературы:**

1. Бёрнс Б. Распределенные системы. Паттерны проектирования. — СПб.: Питер, 2019 — 224 с.: ил. — (Серия «Бестселлеры O’Reilly»). ISBN 978-5-4461-0950-0.
2. Колисниченко Д. Н. Программирование для Android. Самоучитель. — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2020. — 288 с.: ил. ISBN 978-5-9775-6587-5.
3. Рихтер Дж. Clr Via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд. — СПб.: Питер, 2013. — 896 с.: ил. — (Серия «Мастер-класс»). ISBN 978-5-496-00433-6.
4. Mastering PostgreSQL 13 - Packt Publishing, Birmingham, UK, 2020 - 459 c. ISBN 978-1-80056-749-8.
5. Моуэт Э. Использование Docker / пер. с англ. А. В. Снастина; науч. ред. А. А. Маркелов. — М.: ДМК Пресс, 2017. - 354 с.: ил. ISBN 978-5-97060-426-7.