

## **ЭРГОНОМИЧНОСТЬ ИНТЕРФЕЙСА СИСТЕМЫ АДМИНИСТРИРОВАНИЯ ПРЕДПРИЯТИЯ ПО РЕМОНТУ ЦИФРОВОЙ ТЕХНИКИ**

*Аникейченко Д.А., Кобызов Н.С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Агейко О.В. – кандидат технических наук, доцент*

Целью работы является проектирование системы администрирования предприятия по ремонту цифровой техники, способной автоматизировать процесс проведения операций с клиентами.

Для достижения поставленной цели необходимо выполнить следующий ряд задач: – провести обзор существующих систем администрирования предприятия по ремонту цифровой техники; спроектировать архитектуру программного обеспечения с учетом проведенного анализа, спроектировать основные модули системы.

Актуальность разработки системы заключается в том, что на сегодняшний момент существует достаточно много сервисных центров и целесообразно предоставить продукт, который может автоматизировать работу их работу. Данный проект предоставляет возможность упростить заказчику работу с различными операциями на предприятии и повысить качество своей работы.

В настоящее время на предприятии в качестве системы управления заказами и другими данными используются следующие системы: Microsoft Excel; 1С:Предприятие.

Эти системы позволяют хранить необходимые данные, однако их использование становится не совсем удобным, когда значительно увеличивается объем данных. Также в этих системах нелегко поддерживать целостность данных, так как многие данные (например, заказы и клиенты) могут быть связаны друг с другом. И главным минусом этих систем является невозможность работать с ними удаленно через интернет.

Разрабатываемое приложение должно иметь лаконичный и удобный интерфейс, с учетом общих пользовательских требований, опираясь на пожелания заказчика, для быстрой и продуктивной работы.

Приложение должно иметь стандартные элементы управления данными, для простоты его использования. Активности не должны быть перегружены информацией, так как при большом количестве информации на небольшом рабочем пространстве тяжело сосредоточиться на конкретном объекте. Разрабатываемое приложение должно быть понятно новому работнику, который будет использовать систему.

В качестве инструмента для проектирования модели программного средства будет выбран унифицированный язык моделирования – Unified Modeling Language (UML). Это язык для специфицирования, визуализации, конструирования и документирования программных средств.

Составим диаграмму вариантов использования. Варианты использования описывают не только взаимодействия между пользователями и сущностью, но также реакции сущности на получение отдельных сообщений от пользователей и восприятие этих сообщений за пределами сущности. Варианты использования могут включать в себя описание особенностей способов реализации сервиса и различных исключительных ситуаций, таких как корректная обработка ошибок системы. Множество вариантов использования в целом должно определять все возможные стороны ожидаемого поведения системы. Диаграмма вариантов использования представлена на рис. 1.

В системе имеется два типа пользователей: администратор; работник. Работу с приложением будет осуществлять администратор.

Администратор может пройти авторизацию в системе и работать с ней. В обязанности администратора входит работа с данными предприятия – создание и обновление заказов, добавление клиентов в базу и т.д. Также администратор сможет создавать различные отчеты.

Работник не имеет возможности пройти авторизацию, так как работать с приложением может только администратор. Профили работников в приложении используются для того, чтобы в любой момент можно было узнать самую актуальную информацию о заказе – кто из работников над ним работает либо работал.

Разрабатываемая автоматизированная система подразделена на веб-сервер и веб-приложение с пользовательским интерфейсом.

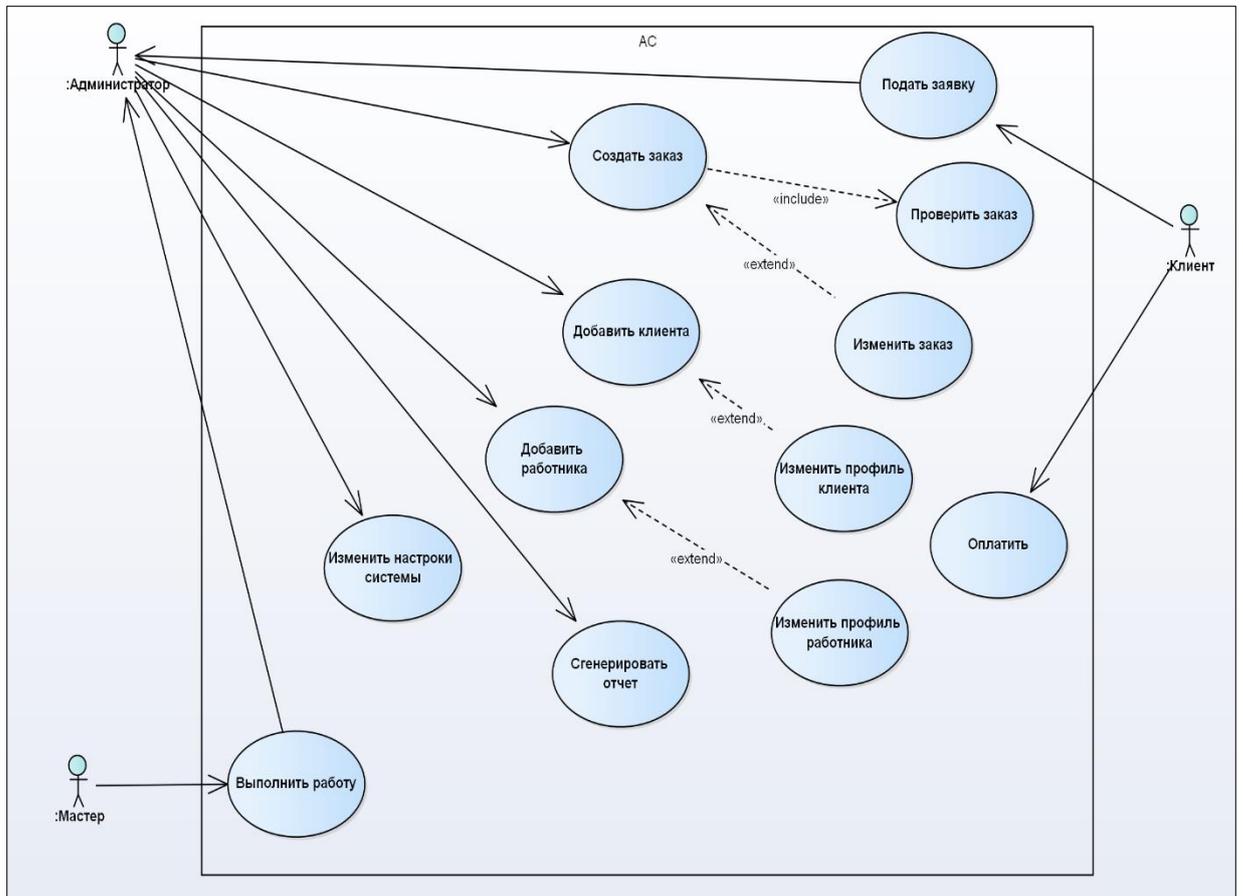


Рисунок 1 – Диаграмма вариантов использования для пользователей

Программный модуль разрабатываемого приложения состоит из двух компонентов: сервер; клиент. Серверная часть программного модуля представляет собой RESTful API. Клиентская часть приложения реализована с помощью фреймворка AngularJS и представляет с собой веб-приложение с пользовательским интерфейсом.

**Список использованных источников:**

1. Эккель, Б. *Философия Java. Библиотека программиста. 4-е изд.* – СПб.: Питер, 2009. – 640 с.
2. Энтони Гонсалвес, *Изучаем Java EE 7 / Энтони Гонсалвес – Изучаем Java EE 7.* – СПб.: Питер, 2014. — 640 с.
3. *Эргономика : учеб. пособие. В 2 ч./ Л. А.Вайнштейн.–Минск : БГУИР, 2018*