

УДК 004

**РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗАЦИИ ПРОЦЕССА
РЕГИСТРАЦИИ, АВТОРИЗАЦИИ И АУТЕНТИФИКАЦИИ В
КОМПЬЮТЕРНОЙ МАСТЕРСКОЙ**

Кытманов А.В., студент гр. ИТб-221, IV курс
Научный руководитель: Ивина О.А., к.н., доцент
Кузбасский государственный технический университет имени Т.Ф.
Горбачева, г. Кемерово

В настоящее время компьютерные мастерские используют различные сервисы для разграничения и упрощения , которое требует точного и актуального учета пользователей, их прав доступа и служебной информации. Традиционные методы учета, основанные на бумажных журналах или электронных таблицах, обладают рядом недостатков: высокая вероятность ошибок при вводе данных, трудоемкость, отсутствие оперативного контроля за выполняемыми действиями сотрудников.

Актуальность разработки автоматизированной системы для компьютерной мастерской обусловлена необходимостью:

- повышения точности учета оборудования пользователей;
- ускорения процесса доступа сотрудника к необходимой рабочей информации;
- минимизации человеческого фактора (сокращение ошибок при ручном заполнении журналов, неправильных записях или назначении ответственных лиц);
- обеспечения прозрачности учета сотрудников.

Одним из ключевых направлений автоматизации является внедрение технологии регистрации, авторизации и аутентификации пользователя, позволяющей быстро автоматизировать процессы проверки прав пользователя.

Процесс учета пользователей в диагностическом центре включает:

- регистрацию пользователя;
- логирование действий;
- присвоение должности пользователя;
- добавление возможности ведения заявок;
- формирование отчетности.

Разрабатываемая система предусматривает следующие функции:

- авторизация и аутентификация пользователя;
- управление списком пользователей и их данными;
- создание учетных записей пользователей;

Диаграмма вариантов использования (Use Case) на рисунке 1 включает три основные роли: менеджер, специалист сервиса и системный администратор. Каждый из акторов обладает определенным уровнем доступа и выполняет набор функциональных операций в системе.

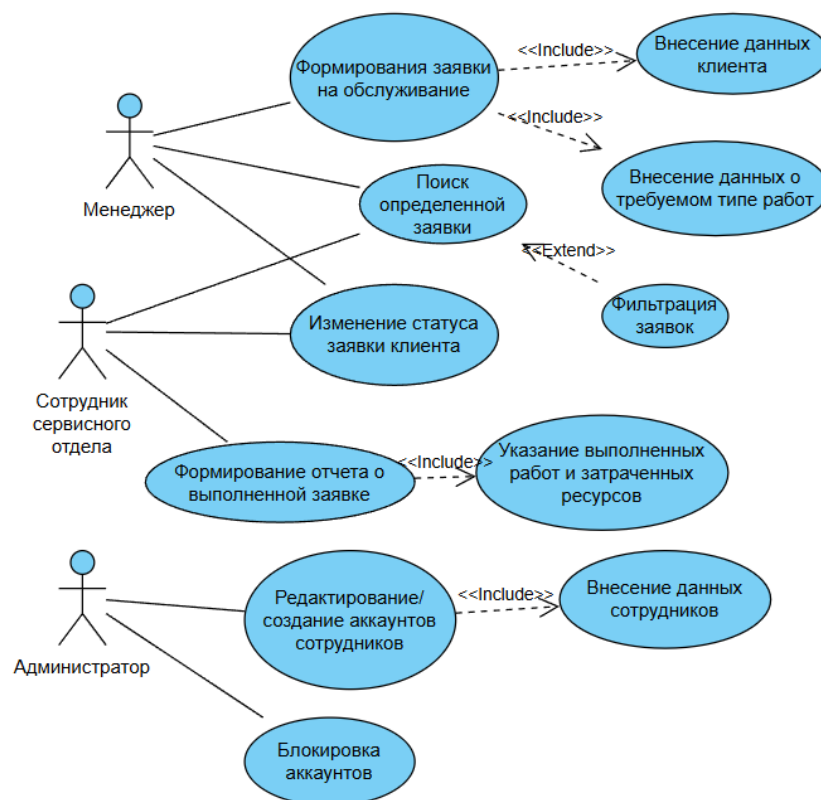


Рисунок 1 - диаграмма вариантов использования системы подачи заявок в сервис

Процесс регистрации пользователя в системе для дальнейшей авторизации и аутентификации отображен на диаграмме деятельности (Рисунок 2). Данный процесс построен на открытии формы, заполнении необходимых данных пользователя и сохранение всех изменений и данных в БД.

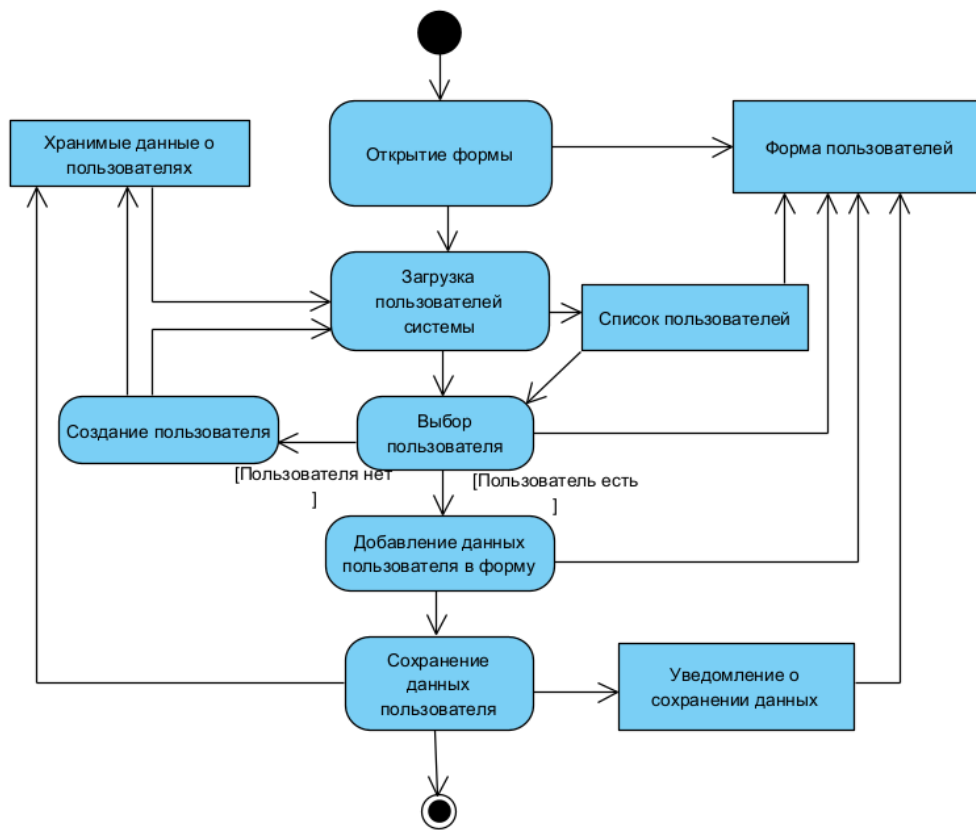


Рисунок 2 - диаграмма деятельности по принятию заявок сервиса

Одним из ключевых направлений автоматизации является внедрение технологии регистрации, авторизации и аутентификации, позволяющая защитить систему и связанную информацию от несанкционированного доступа посторонних лиц.

Каждой учетной записи пользователя автоматически присваиваются уникальные учетные данные, на основе которых обеспечивается безопасный доступ к системе. Механизм `django.auth` используется для аутентификации сотрудников и разграничения прав доступа к функционалу платформы.

Для управления сессиями и правами используется встроенный фреймворк `Django Auth`, который обеспечивает надежную проверку паролей и защиту данных без необходимости реализации собственных алгоритмов шифрования и хранения. Инструментарий позволяет гибко настраивать группы пользователей и уровни доступа (`Permissions`), формируя безопасную среду для работы с ресурсами системы.

Применение django.auth позволяет сократить время на разработку системы безопасности, минимизировать риски утечки данных и упростить администрирование учетных записей.

Процесс авторизации пользователя в системе реализован программно следующим образом:

```
from django.shortcuts import render, redirect
from django.contrib.auth import authenticate, login
from django.http import HttpResponseRedirect
from .forms import LoginForm

def user_login(request):
    if request.method == 'POST':
        form = LoginForm(request.POST)
        if form.is_valid():
            cd = form.cleaned_data
            # Проверка пользователя в БД
            user = authenticate(request,
                               username=cd['username'],
                               password=cd['password'])
            if user is not None:
                if user.is_active:
                    login(request, user) # Авторизация [13]
                    return HttpResponseRedirect('Успешный вход')
                else:
                    return HttpResponseRedirect('Аккаунт заблокирован')
            else:
                return HttpResponseRedirect('Неверный логин или пароль')
        else:
            form = LoginForm()
    return render(request, 'account/login.html', {'form': form})
```

Проверка данных, введенных пользователем, осуществляется при помощи методов библиотеки django.auth.

Пользователь во вкладке “Войти” при входе вводит данные от аккаунта и нажимает кнопку “Войти”. Система сверяет данные логина и хеш-сумму пароля в БД. Если данные совпадают и проверка пройдена (и дополнительные параметры проверки браузера и заголовков пройдены), то пользователю открывается доступ в рабочую вкладку. В ином случае пользователя уведомляют об ошибке.

Результат работы модуля представлен на рисунке 3 и 4.
Проверка

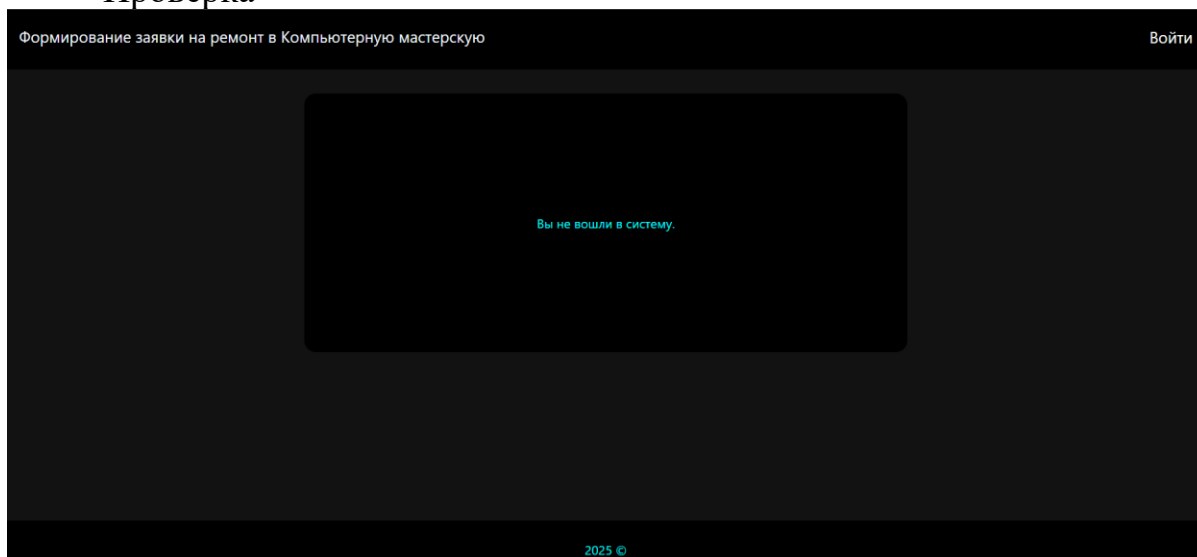


Рисунок 3 – Страница формирования заявки при заходе неавторизованного пользователя

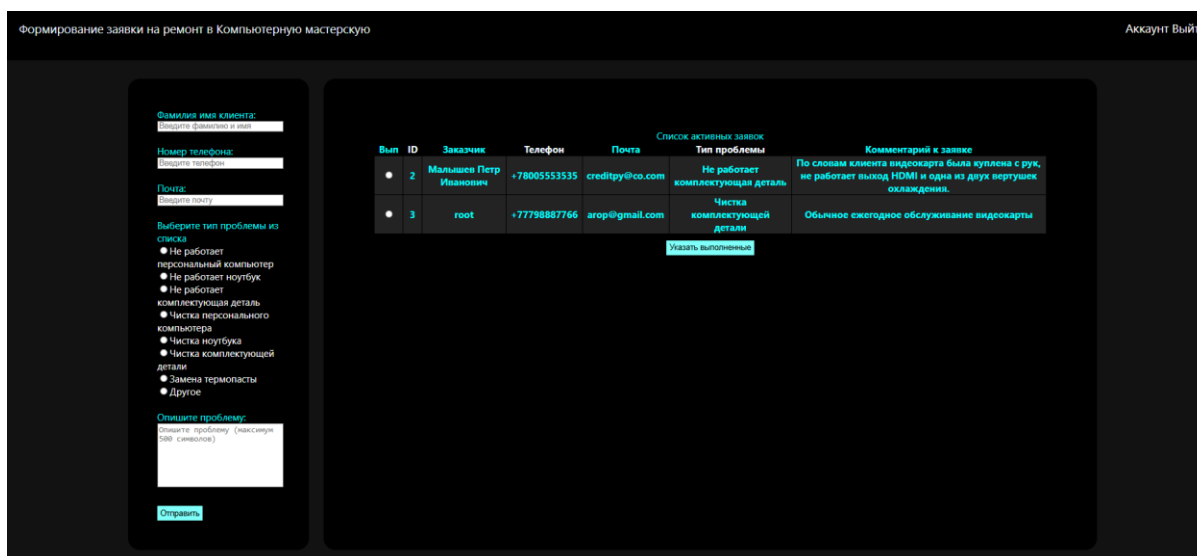


Рисунок 4 – Страница формирования заявки при заходе авторизованного пользователя

Таким образом, библиотека `django.auth` является ключевым компонентом системы, обеспечивающим автоматизацию процесса аутентификации и авторизации.

При разработке системы используются следующие технологии:

- платформа Django Web Framework для создания веб приложения;

- технология HTML, CSS для реализации пользовательского интерфейса;
- система управления базами данных SQLite;
- библиотека django.auth для аутентификации и авторизации.

Диаграмма развертывания представлена на рисунке 4. Она отражает физическую архитектуру системы. Клиентское приложение развернуто на рабочем месте и взаимодействует с сервером базы данных SQLite. Проверка пользователей осуществляется с использованием библиотеки django.auth встроенной в серверное приложение. В качестве получаемых данных при авторизации и аутентификации, поступают данные с клиентского приложения (логин и пароль) для проверки пользователя.

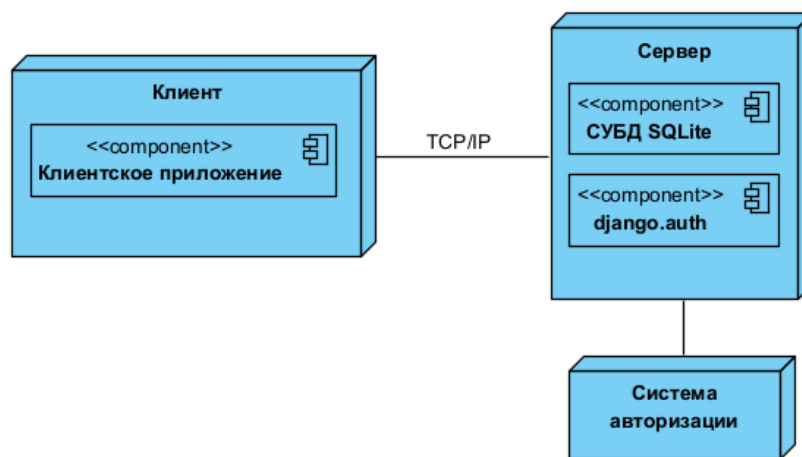


Рисунок 5 – Диаграмма развертывания

Разработка системы автоматизации процесса авторизации и аутентификации в компьютерной мастерской позволяет значительно повысить эффективность пользователей. Особую роль в системе играет использование библиотеки Django.Auth, позволяющей реализовать механизм регистрации, авторизации и аутентификации пользователей системы.

Разработанная система может быть использована в других организациях, где требуется учет и контроль пользовательских учетных записей.

Список литературы:

1. Python – Руководство по Python [Электронный ресурс]. – Режим доступа: <https://metanit.com/python/> (дата обращения: 26.03.2026).
2. Django – Руководство по веб-фреймворку Django [Электронный ресурс]. – Режим доступа: <https://metanit.com/python/django/> (дата обращения: 26.03.2026).
3. Python - программирование на python в примерах и задачах. Автор Васильев А. Н. (дата обращения 27.03.2026)
4. Простой Python. Современный стиль программирования, 2-е издание. Автор: Любанович Б. (дата обращения 27.03.2026)