

Электронный научный журнал «Век качества» ISSN 2500-1841 <http://www.agequal.ru>

2024, №4 [http://www.agequal.ru/pdf/2024/AGE\\_QUALITY\\_4\\_2024.pdf](http://www.agequal.ru/pdf/2024/AGE_QUALITY_4_2024.pdf)

**Ссылка для цитирования этой статьи:**

Барсук И.В., Селезнев А.Ю. Совершенствование системы обслуживания пациентов в поликлинике // Электронный научный журнал «Век качества». 2024. №4. С. 129-149. Режим доступа: <http://www.agequal.ru/pdf/2024/424007.pdf> (доступ свободный). Загл. с экрана. Яз. рус., англ.

УДК 65.015.24

**Совершенствование системы обслуживания пациентов в поликлинике**

**Барсук Игорь Вадимович,**  
*кандидат технических наук, доцент,*  
*кафедра «Интеллектуальные системы в управлении и автоматизации»,*  
*ФГБОУ ВО «Московский технический университет связи и информатики»*  
*111024, город Москва, Авиамоторная ул., д.8а*  
*[igor.v.barsuk@yandex.ru](mailto:igor.v.barsuk@yandex.ru)*

**Селезнев Андрей Юрьевич,**  
*магистрант,*  
*кафедра «Интеллектуальные системы в управлении и автоматизации»,*  
*ФГБОУ ВО «Московский технический университет связи и информатики»*  
*111024, город Москва, Авиамоторная ул., д.8а*  
*[andrewselan2001@gmail.com](mailto:andrewselan2001@gmail.com)*

Анализ современных автоматизированных систем управления медицинскими учреждениями показывает, что в них отсутствует модуль распределения нагрузки между врачами, что может приводить к перегрузке и снижению эффективности работы врачей. Предлагаемая усовершенствованная автоматизированная система управления процессом обслуживания пациентов в поликлинике состоит из двух основных частей. Первая автоматизирует стандартные бизнес-процессы в медицинском учреждении, вторая - модуль с нечеткой моделью - отвечает за распределение нагрузки между врачами. Каждый раз при оформлении записи или отмене записи к врачу происходит пересчет загруженности сотрудника медицинского учреждения. В разработанной системе реализованы возможности клиентов записаться на прием самостоятельно или оформить запись через врача и личный кабинет, доступный как клиентам, так и врачам, просмотреть истории своих посещений; а врачи, кроме просмотра истории, могут выставлять диагнозы, записывать пациентов на прием к другим специалистам. Кроме того, имеется возможность реализовать отдельную базу данных для нечеткой модели, например, для сбора статистики обращений к модели.

**Ключевые слова:** поликлиника; врач; пациент; обслуживание; процесс обслуживания, автоматизированная система управления; нечеткая модель.

## Постановка задачи

Одной из задач улучшения организации работы медицинских учреждений в России является уменьшение очереди пациентов на прием к врачу и обеспечение времени контакта пациента со специалистом, достаточного для постановки правильного диагноза.

Для эффективного управления медицинскими учреждениями применяются автоматизированные системы управления (АСУ МУ), которые должны отвечать следующим требованиям:

- позволять медицинскому персоналу поликлиники управлять информацией о пациентах, включая личные данные, медицинскую историю, управлять расписанием приемов пациентов, записывать новых пациентов на прием, изменять или отменять существующие записи;
- позволять клиентам осуществлять запись на прием к врачу, изменять или отменять существующие записи;
- обеспечивать безопасную авторизацию и регистрацию пользователей, которая гарантирует конфиденциальность медицинской информации и защиту от несанкционированного доступа к данным;
- предоставлять различные уровни доступа для разных пользователей в зависимости от их роли и обязанностей;
- давать возможность просмотра истории посещений, диагнозов;
- предоставлять возможность восстановления пароля, осуществления проверки почты пользователя после регистрации;
- предоставлять доступ к информации о медицинском персонале поликлиники.

Анализ современных АСУ МУ указывает на отсутствие в них системы распределения нагрузки между врачами. Когда в одном медицинском учреждении работает несколько врачей, обслуживающих большое количество пациентов, может возникать ситуация, при которой одни врачи будут перегружены работой, а другие будут свободны. Это может привести к несправедливому распределению

рабочей нагрузки и ухудшению качества оказываемой медицинской помощи. Врачи, перегруженные работой, могут столкнуться с более высоким уровнем стресса и более высокой вероятностью ошибок при обслуживании пациентов.

Увеличить функционал АСУ МУ возможно путем разработки модуля с нечеткой моделью, которая будет распределять нагрузку между врачами. В случае ошибки в расчетах модели вся АСУ МУ перестанет функционировать. Для предотвращения подобной ситуации необходимо вынести нечеткую модель в отдельный программный интерфейс (API). Для взаимодействия этих двух систем можно использовать сетевой протокол прикладного уровня – HTTP. Требования к нечеткой модели, распределяющей нагрузку между врачами, могут быть сформированы следующим образом:

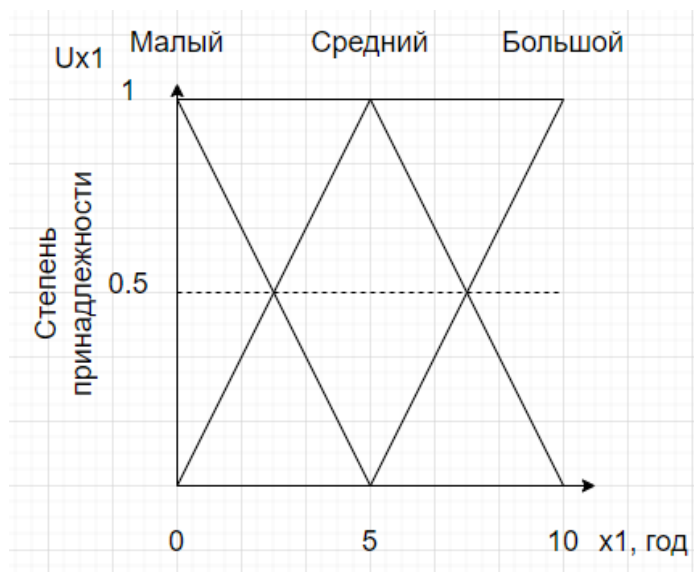
- необходимо реализовать авторизацию для нечеткой модели через JWT (JSON Web Token) токены;
- необходимо реализовать программный интерфейс для нечеткой модели, через который должно происходить взаимодействие с основной системой;
- нечеткая модель должна определять загруженность врача по количеству записей и опыту работы врача.

### **Проектирование нечеткой модели для создания системы рекомендаций, распределяющей нагрузку между врачами**

Нечеткие модели определяют степень принадлежности объекта к классу. В контексте распределения нагрузки между сотрудниками поликлиники нечеткие модели используются для определения оптимального распределения нагрузки между врачами с учетом не только количественных, но и качественных параметров, таких как специализация, квалификация, опыт и занятость медицинского персонала, что позволяет оптимизировать расписание таким образом, чтобы максимизировать количество пациентов, обслуживаемых сотрудниками поликлиники в единицу времени. При построении нечеткой модели, производится операция фаззификации входных величин - перевод реальных

значений в значения логико-лингвистических переменных [1]. Далее необходимо определить моделируемый процесс со следующими характеристиками: входной параметр  $x_1$  – опыт работы; единица измерения – годы; «Малый», «Средний», «Большой»; область применения –  $[0, \infty]$ .

Графическое отображение первого входного параметра приведено на рис. 1.



Составлено авторами

Рис. 1. Графическое отображение первого входного параметра

Функция принадлежности может быть записана в виде формулы (1):

$$U(x) = \omega \cdot \left( \frac{a-|x-e|}{a} \right). \quad (1)$$

Тогда аналитические выражения будут выражаться формулами (2-4):

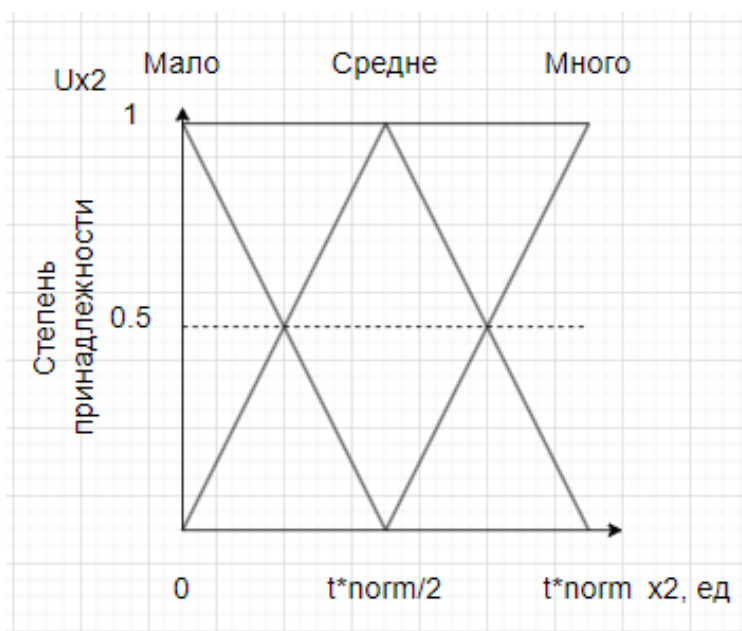
$$\text{Для термина "Малый"} \quad \omega_1 \cdot \left( \frac{5-|x-0|}{5} \right), \text{ где } \omega_1 = \begin{cases} 0 \leq x \leq 5 \\ 0, \text{ else} \end{cases}. \quad (2)$$

$$\text{Для термина "Средний"} \quad \omega_2 \cdot \left( \frac{5-|x-5|}{5} \right), \text{ где } \omega_2 = \begin{cases} 0 \leq x \leq 10 \\ 0, \text{ else} \end{cases}. \quad (3)$$

$$\text{Для термина "Большой"} \quad \omega_3 \cdot \left( \frac{5-|x-10|}{5} \right), \text{ где } \omega_3 = \begin{cases} 5 \leq x \leq 10 \\ 0, \text{ else} \end{cases}. \quad (4)$$

Входной параметр  $x_2$  – количество пациентов, ед; “Мало”, “Средне”, “Много”; область определения –  $[0, t \cdot \text{norm}]$ , где  $t$  – время работы в часах,  $\text{norm}$  – норма пациентов в час.

Графическое отображение второго входного параметра приведено на рис. 2.



Составлено авторами

Рис. 2. Графическое отображение второго входного параметра

Функция принадлежности может быть записана в виде формулы (1).

Тогда аналитические выражения будут описываться формулами (5-7):

$$\text{Для терма "Мало"} \quad \omega_1 \cdot \left( \frac{t \cdot \text{norm}/2 - |x_2 - 0|}{t \cdot \text{norm}/2} \right), \quad (5)$$

$$\text{где } \omega_1 = \begin{cases} 0 \leq x_2 \leq t \cdot \text{norm}/2 \\ 0, \text{ else} \end{cases}.$$

$$\text{Для терма "Средне"} \quad \omega_2 \cdot \left( \frac{t \cdot \text{norm}/2 - |x_2 - t \cdot \text{norm}/2|}{t \cdot \text{norm}/2} \right), \quad (6)$$

$$\text{где } \omega_2 = \begin{cases} t \cdot \text{norm}/2 \leq x_2 \leq t \cdot \text{norm} \\ 0, \text{ else} \end{cases}.$$

$$\text{Для терма "Много"} \quad \omega_3 \cdot \left( \frac{t \cdot \text{norm}/2 - |x_2 - t \cdot \text{norm}|}{t \cdot \text{norm}/2} \right), \quad (7)$$

$$\text{где } \omega_3 = \begin{cases} 0 \leq x_2 \leq t \cdot \text{norm} \\ 0, \text{ else} \end{cases}.$$

Выходной параметр  $y$  – загруженность врача.

Единица измерения –  $y_e$  (условная единица).

Область определения –  $[0, 1]$ : 0 – низкая, 0,5 – средняя, 1 – высокая.

Графическое отображение выходного параметра представлено на рис. 3.



Составлено авторами

Рис. 3. Графическое отображение выходного параметра

Функция принадлежности может быть записана в виде формулы (1).

Тогда аналитические выражения будут описываться формулами (8-10):

$$\text{Для терма "Низкая"} \quad \omega_1 \cdot \left( \frac{0.5 - |x - 0|}{0.5} \right), \text{ где } \omega_1 = \begin{cases} 0 \leq x \leq 0.5 \\ 0, \text{ else} \end{cases} \quad (8)$$

$$\text{Для терма "Средняя"} \quad \omega_2 \cdot \left( \frac{0.5 - |x - 0.5|}{0.5} \right), \text{ где } \omega_2 = \begin{cases} 0 \leq x \leq 1 \\ 0, \text{ else} \end{cases} \quad (9)$$

$$\text{Для терма "Высокая"} \quad \omega_3 \cdot \left( \frac{0.5 - |x - 1|}{0.5} \right), \text{ где } \omega_3 = \begin{cases} 0.5 \leq x \leq 1 \\ 0, \text{ else} \end{cases} \quad (10)$$

Затем проводится расчет значений функции принадлежности.

Если  $x_1 = 7$ ,  $x_2 = 5$ ,  $x_1 = \{A_1, A_2, A_3\}$  и  $x_2 = \{B_1, B_2, B_3\}$ , то итогом процедуры фаззификации будет:  $\mu_{A_1}(x_1^*) = 0$ ;  $\mu_{A_2}(x_1^*) = 0,6$ ;  $\mu_{A_3}(x_1^*) = 0,4$ ;  $\mu_{B_1}(x_2^*) = 0,75$ ;  $\mu_{B_2}(x_2^*) = 0,25$ ;  $\mu_{B_3}(x_2^*) = 0$ .

Далее составляется база правил для нечеткой модели, определяется минимально необходимое количество правил в модели по формуле (11).

$$R = Z^W, \quad (11)$$

где  $R$  – количество правил в базе,  $Z$  – количество термов (нечетких множеств) на каждом входе,  $W$  – количество входов в модели.

Затем нужно провести расчет:

- R1: Если  $(X1 = A1) [0]$  и  $(X2 = B1) [0,75]$ , то  $(y = C1) = 0$ ;  
R2: Если  $(X1 = A1) [0]$  и  $(X2 = B2) [0,25]$ , то  $(y = C2) = 0$ ;  
R3: Если  $(X1 = A1) [0]$  и  $(X2 = B3) [0]$ , то  $(y = C3) = 0$ ;  
R4: Если  $(X1 = A2) [0,6]$  и  $(X2 = B1) [0,75]$ , то  $(y = C1) = 0,45$ ;  
R5: Если  $(X1 = A2) [0,6]$  и  $(X2 = B2) [0,25]$ , то  $(y = C2) = 0,15$ ;  
R6: Если  $(X1 = A2) [0,6]$  и  $(X2 = B3) [0]$ , то  $(y = C3) = 0$ ;  
R7: Если  $(X1 = A3) [0,4]$  и  $(X2 = B1) [0,75]$ , то  $(y = C1) = 0,3$ ;  
R8: Если  $(X1 = A3) [0,4]$  и  $(X2 = B2) [0,25]$ , то  $(y = C2) = 0,1$ ;  
R9: Если  $(X1 = A3) [0,4]$  и  $(X2 = B3) [0]$ , то  $(y = C2) = 0$ .

После этого строится графически результирующая функция принадлежности для выходного параметра, результат изображен на рис. 4.



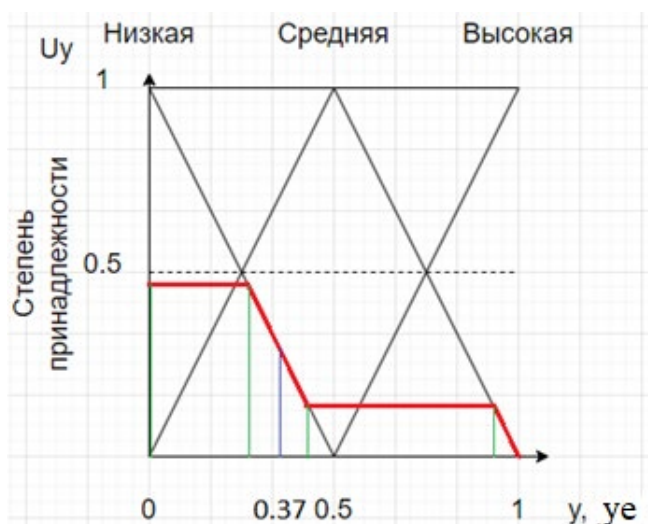
Составлено авторами

Рис. 4. Результирующая функция принадлежности для выходного параметра

Потом проводится дефаззификация. Под дефаззификацией нечеткого множества, являющегося результатом вывода, понимается операция нахождения четкого значения, которое наиболее «рациональным» образом представляло бы это множество [2]. Для этого используют следующие методы: метод среднего максимума (Middle of Maxima, MM); метод центра тяжести (Center of Gravity, CG);

метод высот (Height, H). В заключение сравниваются полученные с помощью данных методов результаты, выбирается наиболее подходящий метод.

Результат, полученный с помощью метода CG, представлен на рис. 5. Вычисления для метода CG осуществляются по формуле (12), пределы интегрирования задаются областью определения  $y$ . Для решаемых задач результат, полученный в данном примере, является неадекватным, т.к. в итоге получается, что опытный сотрудник, рабочий день которого равен 8-ми часам, при 5-ти пациентах классифицируется как средне занятый. Поэтому данный метод дефаззификации не подходит для разрабатываемой автоматизированной системы управления.



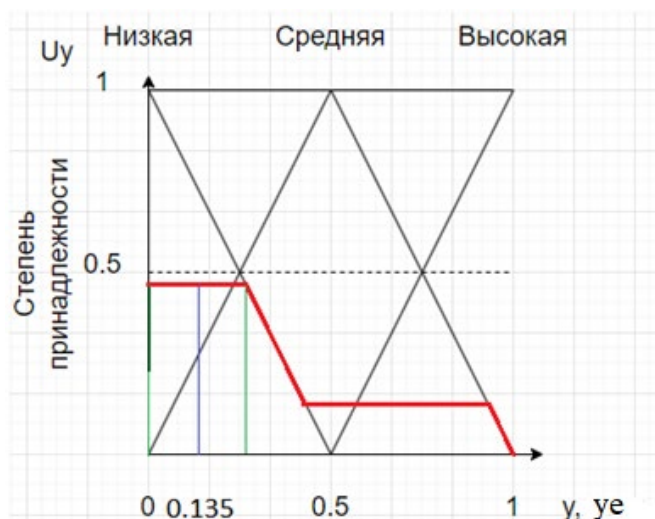
Составлено авторами

Рис. 5. Результат, полученный с помощью метода CG

$$y^* = \frac{\int y U_{res}(y) dy}{\int U_{res}(y) dy} \quad (12)$$

Результат, полученный с помощью метода ММ, представлен на рис. 6.



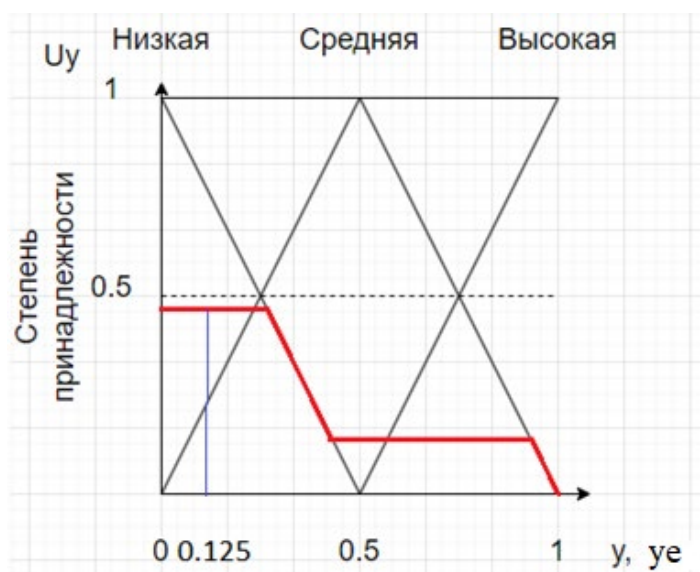


Составлено авторами

Рис. 6. Результат, полученный с помощью метода ММ

Результат, полученный с помощью метода Н, представлен на рис. 7.

Вычисления по методу Н осуществляются по формуле (13).



Составлено авторами

Рис. 7. Результат, полученный с помощью метода Н

$$y^* = \frac{\sum_{j=1}^m y_j UB_j(y)}{\sum_{j=1}^m UB_j(y)}, \text{ где } m - \text{число правил.} \quad (13)$$

Задача классификации состоит в предсказании категории объектов и их разделении по определенным и заданным заранее признакам [3]. Для решения данной задачи наилучшим образом подходят метод среднего максимума и метод высот.

Достоинства метода среднего максимума состоят в простоте реализации и низкой стоимости вычислений, недостатками являются низкая чувствительность с точки зрения нечеткой модели. Фактически учитывается только то результирующее нечеткое множество, которое имеет наибольшую высоту, то есть на итоговое значение выходного параметра влияет только одно правило.

Достоинствами метода высот являются простота реализации, низкая стоимость вычислений. Этот метод более чувствителен к изменению входных сигналов, чем метод среднего максимума, так как в дефаззификации участвуют все активизированные функции принадлежности заключений (все активные правила).

### **Проектирование базы данных и функциональная схема работы проектируемой АСУ МУ**

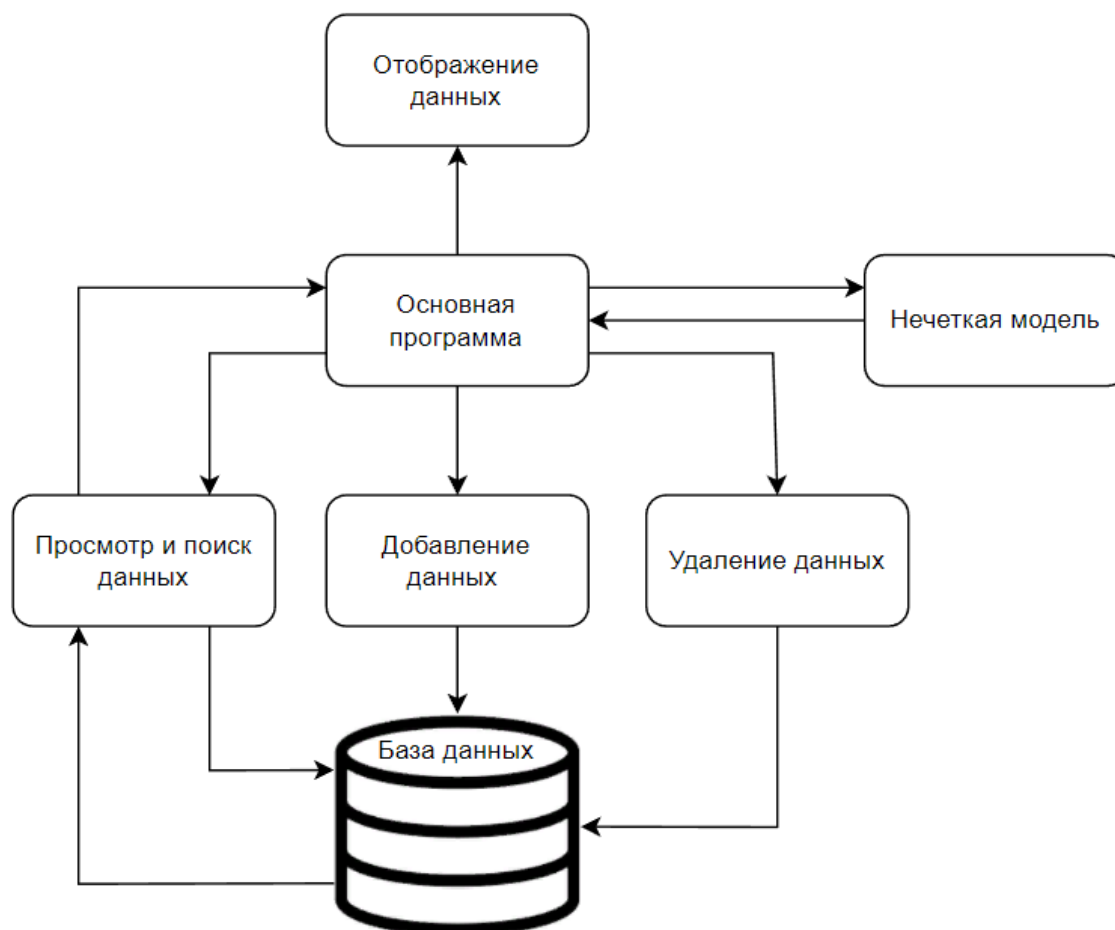
На начальном этапе проектирования баз данных проводится инфологическое моделирование, его результаты используются на заключительном даталогическом этапе, когда решается проблема конструирования непосредственно базы данных в среде конкретной СУБД [4]. Этапы составления инфологической модели:

1. Определение всех сущностей, которые будут храниться в базе данных.
2. Определение атрибутов. Атрибуты – это свойства сущностей, которые будут храниться в базе данных.
3. Определение связей между сущностями.
4. Создание диаграммы, которая отображает структуру базы данных.

В АСУ МУ выделены следующие сущности: 1) assigned\_diagnoses (назначенные диагнозы); 2) client (клиент); 3) clientcheck (запросы на запись пациентов которые оформляются медицинским сотрудником); 4) diagnoses (список болезней); 5) list of visits (список посещений врачей); 6) loading doctors (уровень загрузки врачей); 7) loading levels (уровни загрузки врачей); 8) news (новости); 9) position (должности сотрудников поликлиники); 10) schedule (график работы сотрудников); 11) staff (сотрудники); 12) visitcheck (проверка возможности оформить запись пациента к врачу); 13) auth group (группы пользователей); 14) auth

group permissions (права групп пользователей); 15) auth permission (права пользователей); 16) auth user (пользователи); 17) auth user groups (назначенные пользователям группы); 18) auth user user permissions (назначенные пользователям права); 19) django admin log (логирование админ-панели); 20) django content type (механизм, связывающий модели с системой авторизации и уровнями доступа); 21) django migrations (миграции); 22) django session (сессии).

На основе приведенных выше сущностей построена инфологическая модель, трансформированная с учетом требований и ограничений конкретной СУБД [5]. Исходя из приведённых выше отношений, построена схема получившейся БД. Функциональная схема работы проектируемой АСУ МУ представлена на рис. 8.



Составлено авторами

Рис. 8. Функциональная схема работы проектируемой АСУ МУ

АСУ МУ решает четыре основные задачи: добавление данных в базу данных; удаление данных из базы данных; просмотр и поиск данных; отображение информации, полученной из базы данных. Изображенная архитектура называется MVC (Model–View–Controller). Структура архитектуры MVC разделяет приложение на три основные группы компонентов: модели, представления и контроллеры. Это позволяет реализовать принципы разделения задач [6]. Модель (Model) представляет данные, с которыми работает приложение, и содержит логику работы с ними.

### **Реализация усовершенствованной АСУ МУ**

Для реализации разработанной АСУ МУ используются Django и sqllite. Django - это бесплатный и открытый фреймворк для создания программ на языке Python [7]. Он обладает широким набором инструментов и функций, позволяющих ускорить и упростить процесс разработки [8].

SQLite - это библиотека на языке Си, которая реализует небольшой, быстрый, автономный, высоконадежный и полнофункциональный движок базы данных SQL [9]. SQLite может использоваться как база данных по умолчанию в Django. Использование SQLite в Django зависит от размера проекта и специфических требований. Если разрабатывается маленький или средний проект, SQLite может быть хорошим выбором. Если проект большой и требует высокой производительности, то лучше использовать другую СУБД, такую как PostgreSQL.

Нечеткая модель для создания системы рекомендаций, распределяющей нагрузку между врачами, реализована с помощью 4-х функций без использования сторонних библиотек языка Python [10].

*Первая функция* (solveInputValue) принимает семь параметров: опыт работы (тип данных - int); количество пациентов, записавшихся на прием в данный момент (тип данных - int); три границы, используемые для расчета области определения опыта работы (тип данных - int); нормальное количество пациентов в час (тип данных - int); длительность рабочего дня в часах (тип данных - int). После этого

производится расчет значений функции принадлежности для первого параметра  $x_1$  (опыт работы) и для второго параметра  $x_2$  (количество пациентов). Функция `solveInputValue` возвращает массив из шести значений функции принадлежности.

*Вторая функция* (`ruleBase`) принимает массив из 6-ти значений функции принадлежности, полученный в качестве результата работы функции `solveInputValue`, и строит из этого массива базу правил для нечеткой модели.

*Третья функция* (`Height`) принимает базу правил, полученную в качестве результата работы функции `ruleBase`, и осуществляет дефаззификацию с помощью метода высот. Алгоритм распределения значений по трем массивам, на основе которых подсчитывается сумма высот, представлен в [10].

*Четвертая функция* (`out_class`) осуществляет подсчет итоговых результатов, вывод необходимой информации в консоль.

Пример расчета с помощью описанных функций показан на рис. 9.

```
0 0.6 0.4
Кол-во возможных посещений: 40
0.75 0.25 0
-----
Height
0 0.6 0.4
Кол-во возможных посещений: 40
0.75 0.25 0
Степень принадлежности к терму 'Низкая загрузка': 0.75
Степень принадлежности к терму 'Средняя загрузка': 0.25
Степень принадлежности к терму 'Высокая загрузка': 0
('C1', 0.125)
PS C:\Users\andrey> █
```

Составлено авторами

Рис. 9. Пример расчета уровня загруженности медицинского персонала с помощью нечеткой модели: C1 – сотрудник имеет низкую загруженность; C2 – сотрудник имеет среднюю загруженность; C3 – сотрудник имеет высокую загруженность

Интерфейсом для нечеткой модели служит отдельный модуль Application Programming Interface (API), который используется для взаимодействия программ или сервисов друг с другом [11]. Для реализации API используется Django Rest Framework (DRF) - это мощный и гибкий инструмент для создания веб-API [12]. DRF расширяет возможности Django и добавляет дополнительные функции,

такие как авторизация и аутентификация, сериализация и десериализация данных, работа с различными форматами данных, включая JSON.

Для реализации базы данных АСУ МУ используется Django ORM. ORM - это набор классов, добавляющих еще один уровень абстракции к таблицам, хранящимся в базе данных. ORM автоматически генерирует SQL-запросы и выполняет их на стороне базы данных. ORM-фреймворки также обеспечивают автоматическую проверку целостности данных и миграцию схемы базы данных. На основе даталогической модели необходимо реализовать двенадцать таблиц, остальные сгенерируются фреймворком Django автоматически.

Таблица, в которой хранятся новости медицинской организации, содержит следующие атрибуты: 1 - название новости; 2 - текст новости; 3 - уникальный текстовый идентификатор для новости; 4 - фотография, прикрепленная к новости; 5 - время создания новости; 6 - время изменения новости; 7 - опубликована новость или нет; 8 – наименование пользователя, создавшего и опубликовавшего новость.

Таблица, которая хранит информацию о сотрудниках поликлиники, содержит следующие атрибуты: 1 - уникальный текстовый id; 2 – ФИО; 3 - дата рождения; 4 – фото; 5 – паспорт; 6 – номер телефона; 7 – адрес; 8 - дата приема на работу; 9 - уровень образования; 10 - образовательное учреждение; 11 - год окончания образования; 12 - специальность по диплому; 13 – должность; 14 – пользователь; 15 - опыт работы (в годах); 16 - место работы.

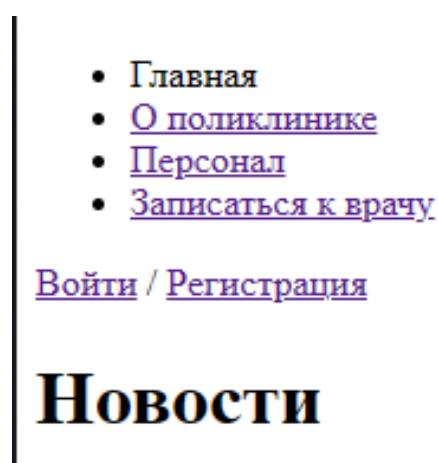
Таблица, которая хранит информацию об уровнях загрузки врачей по дням, не только содержит следующие атрибуты: 1 – сотрудник; 2 – дата; 3 – загрузка, но и название уровня загрузки.

В таблицах с информацией о должностях сотрудников поликлиники имеются атрибуты: 1 - заработная плата; 2 – должность; 3 - уникальный текстовый id; 4 - нормальное количество пациентов в час; 5 – врач; в таблицах с информацией о клиентах поликлиники: 1 – ФИО; 2 – адрес; 3 - дата рождения; 4 - номер телефона; 5 - уникальный текстовый id; 6 – пол; 7 – пользователь.

В таблицах с информацией о записях к врачу, которые оформляются медицинским сотрудником, отражаются такие атрибуты, как: 1 – пользователь; 2 – дата; 3 – время; 4 – сотрудник, а также: 1 - дата посещения; 2 - подтверждение записи; 3 – клиент; 4 – врач. В таблице с информацией о расписании сотрудников поликлиники хранятся следующие данные: 1 - день недели; 2 - время начала работы в день; 3 – время окончания работы в день; 4 – сотрудник.

Таблица, которая хранит информацию о возможных диагнозах, содержит название болезни; о поставленных диагнозах: 1 - посещение; 2 – диагноз; 3 - комментарий врача; о возможностях клиентов записаться на прием к врачу: 1 – клиент; 2 – должность.

Усовершенствованные бизнес-логика и пользовательский интерфейс разработаны в соответствии с требованиями, предъявляемыми к АСУ МУ. Главный экран системы с основными вкладками изображен на рис. 10.



Составлено авторами

Рис. 10. Главный экран системы, на котором представлены все основные вкладки

Для просмотра медицинского персонала поликлиники нужно выбрать вкладку «Персонал». При необходимости имеется возможность отфильтровать список по специализации. Далее для работы с системой необходимо авторизоваться, зарегистрироваться. Для этого созданы формы авторизации и регистрации с подтверждением почты. При успешном заполнении формы регистрации на почту отправляется письмо с одноразовой ссылкой для

подтверждения регистрации. При вводе данных осуществляется проверка, например, пароль не может быть короче 8 символов, пароль не должен быть похож на адрес почты и т.д. Если пользователь забыл пароль, у него есть возможность восстановить его с помощью почты. Пользователю отправляется такое же письмо, как при регистрации, с одноразовой ссылкой, после чего он попадает на форму восстановления пароля. Затем нужно осуществить вход в систему. Для этого можно воспользоваться уже созданным ранее аккаунтом.

После успешной авторизации пользователь возвращается на главную страницу. Главная страница после авторизации меняется, теперь у пользователя появляется возможность выйти из системы и перейти в личный кабинет.

У пользователя есть возможность просмотреть на соответствующей вкладке историю посещений, при необходимости отменить запись к врачу. Для записи на прием нужно перейти на вкладку «Записаться к врачу», далее выбрать специализацию врача. После этого отобразится календарь с возможностью выбрать день для записи, список доступных врачей с уровнями загруженности, отсортированных по времени, на которое осуществляется запись. Загруженность врача нечеткая модель считает по данным, которые отправляются основной системой при каждом оформлении записи к врачу, при каждой отмене записи к врачу.

При попытке пациента записаться на прием ему будет сначала предлагаться записаться к врачу с меньшей загруженностью (рис. 11). Система также предусматривает, что с ней может работать врач, администратор.



[Эндокринолог](#)

[Хирург](#)

[Окулист](#)

**Месяц Май 2023**

Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2	3	4	5	6	7
8	<a href="#">9</a>	<a href="#">10</a>	<a href="#">11</a>	<a href="#">12</a>	<a href="#">13</a>	<a href="#">14</a>
<a href="#">15</a>	<a href="#">16</a>	<a href="#">17</a>	<a href="#">18</a>	<a href="#">19</a>	<a href="#">20</a>	<a href="#">21</a>
<a href="#">22</a>	<a href="#">23</a>	<a href="#">24</a>	<a href="#">25</a>	<a href="#">26</a>	<a href="#">27</a>	<a href="#">28</a>
<a href="#">29</a>	<a href="#">30</a>	<a href="#">31</a>				

[Предыдущий месяц](#) [Следующий месяц](#)

May 31, 2023, 10 a.m.

Когут Оксана Александровна

Терапевт

Загруженность: 1



19

[Записаться на прием](#)

Составлено авторами

Рис. 11. Оформление записи к врачу

Для того, чтобы авторизоваться в системе под аккаунтом врача, можно воспользоваться уже созданным ранее аккаунтом. Врач через личный кабинет может просматривать информацию о записях. При открытии какой-либо записи выводится история диагнозов, поставленных врачами с той же специальностью, имеется возможность поставить новый диагноз. Форма для создания нового диагноза доступна только врачам.

Также врач имеет возможность записать клиента на прием. Для этого врач переходит по ссылке «Создать новую запись», которая находится в личном

кабинете. Далее выбирает нужного пациента, затем повторяется процесс записи на прием, описанный выше (рис. 11).

Для администрирования данной системы предусмотрена панель администратора. Главный экран панели администратора представлен на рис. 12. Доступ к панели администратора есть только у администратора.

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

CLINICWEBSITE		
График работы	+ Add	Change
Диагнозы	+ Add	Change
Должности	+ Add	Change
Загрузка врачей	+ Add	Change
Клиенты	+ Add	Change
Новости	+ Add	Change
Персонал	+ Add	Change
Поставленный диагнозы	+ Add	Change
Список посещений	+ Add	Change
Уровни загрузки	+ Add	Change
Уровни клиента	+ Add	Change

Составлено авторами

Рис. 12. Главный экран панели администратора

При необходимости имеется возможность редактирования, добавления, удаления информации для любой таблицы базы данных через админ-панель.

Результаты тестирования усовершенствованной АСУ МУ показали, что все компоненты системы функционируют без сбоев и ошибок, программа достаточно проста и понятна для пользователей, удобна для администрирования, имеет дружелюбный пользовательский интерфейс и на данный момент не нуждается в каких-либо дополнениях.

## Выводы

Существующее значительное количество автоматизированных систем управления медицинскими учреждениями (АСУ МУ) имеют один важный недостаток - отсутствие в них модуля распределения нагрузки между врачами, что приводит к перегрузке и снижению эффективности работы врачей, увеличению очереди на прием к врачу.

В целях совершенствования существующей АСУ МУ определены требования, которым должна отвечать система, предназначенная для врачей и пациентов, спроектированы нечеткая модель с двумя входными параметрами, распределяющая нагрузку между сотрудниками поликлиники и база данных для АСУ процессом обслуживания пациентов в поликлинике, состоящая из двадцати двух таблиц. В качестве метода дефазификации был выбран метод высот, который показал наиболее адекватные результаты при классификации врачей на свободных, имеющих среднюю загруженность, загруженных.

Для реализации АСУ МУ выбран язык программирования Python, являющийся одним из наиболее популярных языков и обладающий широким спектром библиотек и инструментов. Кроме того, Python имеет простой и интуитивно понятный синтаксис, что упрощает разработку и поддержку кода.

Усовершенствованная автоматизированная система управления процессом обслуживания пациентов в поликлинике реализована в соответствии с установленными требованиями и функциональной схемой работы АСУ МУ. Проведенное тестирование усовершенствованной АСУ МУ показало, что программа работает без сбоев и ошибок, удобна в эксплуатации для пациентов и врачей, улучшает качество обслуживания. Разработанная АСУ МУ может быть рекомендована к внедрению во всех поликлиниках страны, оборудованных современными техническими средствами информационных технологий.

### Список литературы

1. Технология нечеткой логики [Электронный ресурс] // ДинСофт. - URL: <http://www.dynsoft.ru/fuzzy.php> (дата обращения: 12.05.2023).
2. Пегат, А. Нечеткое моделирование и управление / Пер. с англ. – 2-е изд. - М.: БИНОМ. Лаборатория знаний, 2013. - 798 с.
3. Evergreens [Электронный ресурс]. - URL: <https://evergreens.com.ua/ru/articles/classical-machine-learning.html> (дата обращения: 12.05.2023).
4. Сухомлинов, А.И. Инфологическое моделирование: учебно-методическое пособие / А.И. Сухомлинов. – Владивосток: Издательство Дальневосточного федерального университета, 2021. – 36 с.
5. Мердина, О.Д. Базы данных: учебное пособие. – СПб.: Изд-во СПбГЭУ, 2019. – 99 с.
6. Общие сведения ASP.NET Core MVC [Электронный ресурс] // Microsoft Learn. - URL: <https://learn.microsoft.com/ru-ru/aspnet/core/mvc/overview?view=aspnetcore-7.0> (дата обращения: 11.05.2023).
7. Язык программирования Python [Электронный ресурс] // WEB-creator. - URL: <https://web-creator.ru/articles/python> (дата обращения: 13.05.2023).
8. Djangoproject [Электронный ресурс]. - URL: <https://www.djangoproject.com/> (дата обращения: 13.05.2023).
9. SQLite [Электронный ресурс]. - URL: <https://www.sqlite.org/index.html> (дата обращения: 13.05.2023).
10. Свидетельство о государственной регистрации программы для ЭВМ №20224610719 Российская федерация. Программа реализации нечеткой модели для создания системы рекомендаций, распределяющей нагрузку между врачами: № 20224610719: заявл. 29.12.2023: опубл. 12.01.2024 / И.В. Барсук, А.Ю. Селезнев; заявитель МТУСИ.

11. Что такое API простыми словами [Электронный ресурс] // Яндекс Образование. - URL: <https://academy.yandex.com/journal/chto-takoe-api> (дата обращения: 14.05.2023).
12. Django REST framework» [Электронный ресурс] - URL: <https://www.django-rest-framework.org/> (дата обращения: 14.05.2023).

### **Improving the patient care system in the polyclinic**

***Barsuk Igor Vadimovich,***  
*Ph.D., Associate Professor,*  
*Department of "Intelligent systems in Management and Automation",*  
*Moscow Technical University of Communications and Informatics,*  
*8a Aviamotornaya str., Moscow, 111024*  
*igor.v.barsuk@yandex.ru*

***Seleznev Andrey Yurievich,***  
*Master,*  
*Department of "Intelligent systems in Management and Automation"*  
*Moscow Technical University of Communications and Informatics*  
*8a Aviamotornaya str., Moscow, 111024*  
*andrewselan2001@gmail.com*

The analysis of modern automated management systems of medical institutions shows that they lack a module of load distribution between doctors, which can lead to overload and reduce the efficiency of doctors' work. The proposed advanced automated management system of the patient care process in a polyclinic consists of two main parts. The main one automates standard business processes in a medical institution, and the second one, a module with a fuzzy model, is responsible for load distribution between doctors. Every time a record is made or a doctor's appointment is canceled, the workload of an employee of the medical center is recalculated. The developed system realizes the possibility for clients to make an appointment on their own or make an appointment through a doctor and a personal cabinet available to both clients and doctors, to view the history of their visits, and doctors, in addition to viewing the history, can make diagnoses, make appointments for patients to see other specialists. In addition, there is a possibility to realize a separate database for the fuzzy model, for example, to collect statistics of visits to the model.

**Keywords:** polyclinic; doctor; patient; service; service process, automated control system; fuzzy model.