

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ НА ОСНОВЕ
КОГНИТИВНО-ПОВЕДЕНЧЕСКОЙ МОДЕЛИ АВС И ДНЕВНИКА
СМЭР ДЛЯ САМОРЕГУЛЯЦИИ И АНАЛИЗА ЭМОЦИОНАЛЬНЫХ
СОСТОЯНИЙ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета компьютерных наук и информационных технологий
Гниций Екатерины Николаевны

Научный руководитель

доцент, к. ф.-м. н.

Ю. Н. Кондратова

Заведующий кафедрой

к. ф.-м. н., доцент

С. В. Миронов

Саратов 2026

ВВЕДЕНИЕ

Актуальность темы: современный рост интереса к цифровым инструментам, направленным на улучшение эмоционального состояния, связан с увеличением уровня стресса, эмоционального выгорания и тревожных расстройств. Одним из наиболее эффективных и научно обоснованных подходов к стабилизации состояния является когнитивно-поведенческая терапия (КПТ), включающая модель ABC А. Эллиса и дневник СМЭР, применяемый в отечественной психологической практике. Согласно модели ABC, эмоциональные реакции определяются не самим событием, а его интерпретацией; протокол СМЭР формализует эту цепочку в виде последовательности «ситуация — мысли — эмоции — реакция» и удобен для переноса в цифровую форму.

Несмотря на высокую результативность методик, их использование в цифровых продуктах остаётся ограниченным. На рынке мобильных приложений представлено множество трекеров настроения и личных дневников, однако большинство из них ограничивается фиксацией эмодзи или свободным текстом без детального анализа взаимосвязей между событиями, мыслями, эмоциями и реакциями. Существующие решения редко обеспечивают глубокий анализ текста пользователя с применением современных методов обработки естественного языка и нейронных сетей, адаптированных к особенностям русскоязычных записей.

Цель бакалаврской работы — разработка мобильного приложения для саморегуляции эмоциональных состояний на основе модели ABC и дневника СМЭР, интегрированного с нейронной сетью для автоматической классификации эмоционального содержания текста пользователя.

Поставленная цель определила следующие задачи:

1. провести обзор аналогов, применяющих психологические методики поддержки и саморегуляции;
2. выбрать среду разработки и технологии реализации программного комплекса;
3. спроектировать и разработать мобильное приложение «Дневник эмоций»;
4. интегрировать средства NLP-анализа текста для автоматического определения эмоциональной окраски пользовательских записей.

Методологические основы разработки мобильного приложения для саморегуляции и анализа эмоциональных состояний представлены в работах А. Эл-

лиса, А. Бека, Д. Ковпака, П. Данеляна, Н. Ауельбековой, А. Колмогоровой, С. Дворникова, Е. Грибкова, Ю. Ехлакова, Е. Митякова, А. Шмелёвой, А. Ладынина, А. Баланова, Е. Турнецкой, А. Аграновского, Ю. Воронцова, И. Романовой и А. Наместникова.

Теоретическая значимость работы состоит в междисциплинарном сопоставлении методов когнитивно-поведенческой психотерапии и современных средств программной инженерии: показано, как протокол СМЭР и модель ABC могут быть формализованы в структуре мобильного приложения и дополнены автоматическим анализом русскоязычного текста с помощью нейронных сетей.

Практическая значимость бакалаврской работы заключается в создании программного комплекса, объединяющего структурированный дневник СМЭР, справочник из 32 эмоций по Р. Плутчику, локальную нейросетевую подсказку при заполнении записи и опциональный сводный анализ накопленных данных. Реализация такого решения облегчает самонаблюдение между сессиями когнитивно-поведенческой терапии, снижает трудность сопоставления записей с эмоциональными состояниями и делает повседневную работу с эмоциями более системной и доступной.

Краткая характеристика материалов исследования: в работе использованы научные публикации по когнитивно-поведенческой терапии, эмоциональной саморегуляции и обработке русскоязычных текстов; документация средств разработки (React Native, Expo, Node.js, Express, PostgreSQL, Python, Flask); корпус CEDR для дообучения модели RuBERT-tiny2; материалы сравнительного анализа мобильных аналогов (Penzu, Mood Tracker, eMoods); а также результаты функционального и нефункционального тестирования программного комплекса.

Структура и объём работы: бакалаврская работа состоит из введения, двух разделов («Обзор технологий и литературы для разработки мобильного приложения» и «Разработка и реализация мобильного приложения»), заключения, списка использованных источников и пяти приложений. Общий объём работы — 57 страниц, из них 45 страниц — основное содержание (введение, разделы, заключение), 14 рисунков, 8 таблиц; приложения А–Г — фрагменты исходного кода ключевых модулей, приложение Д — flash-носитель с полным исходным кодом проекта; список использованных источников — 22 наименования.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Обзор технологий и литературы для разработки мобильного приложения» посвящён теоретическому обоснованию проекта, анализу предметной области и выбору средств реализации.

Рассмотрены основы когнитивно-поведенческой терапии, рационально-эмоциональная модель ABC А. Эллиса и протокол СМЭР Д. В. Ковпака. Показано соответствие элементов обеих схем: активирующее событие (А) — ситуация (С), убеждения (В) — мысли (М), эмоции и реакция (С) — эмоции с интенсивностью и (Э) реакция тела и поведения (Р). Обе схемы отражают когнитивную цепочку «событие → интерпретация → переживание → действие», однако СМЭР детализирует реакцию на телесный и поведенческий уровни и фиксирует интенсивность эмоций, что важно для последующей аналитики. Такая структура удобна для переноса в мобильный интерфейс, где каждому элементу соответствует отдельное поле формы.

В обзоре литературы проанализированы исследования по эмоциональной саморегуляции и связи эмоционального интеллекта с управлением поведением (П. Данелян, Н. Ауельбекова); особенности сентимент-анализа и распознавания эмоций в русскоязычных текстах, в том числе при работе с эмоционально нейтральными фрагментами (А. Колмогорова, С. Дворников); современные алгоритмы NLP, включая трансформерные архитектуры (Е. Барщевский, Е. Грибков, Ю. Ехлаков); архитектурные подходы к разработке программных систем и микросервисные решения (А. Баланов, Е. Турнецкая, А. Заяц); технологии мобильной разработки и кроссплатформенные фреймворки (Ю. Воронцов); проектирование баз данных и обеспечение согласованности пользовательских данных (И. Романова, А. Наместников, Р. Мамедли). Обзор подтвердил, что модели ABC и СМЭР формализуются для цифровых инструментов, а современные NLP-методы позволяют анализировать эмоционально окрашенные пользовательские тексты при адаптации к особенностям русского языка.

Выполнен сравнительный анализ мобильных аналогов Penzu, Mood Tracker — Diary, Journal и eMoods Wellness Tracker. Penzu предоставляет свободный личный дневник без структуры КПТ: отсутствуют поля СМЭР, шкала интенсивности эмоций, аналитика динамики состояний и подбор эмоций по тексту с помощью нейросети. Mood Tracker ориентирован на выбор «настроения» из ограниченного набора эмодзи (около 17 меток) и кратких заметок; пользователь

не фиксирует отдельно ситуацию, автоматические мысли, телесные и поведенческие реакции, шкала интенсивности 1–10 отсутствует; из сильных сторон — графики и статистика по журналу. eMoods Wellness Tracker предлагает базовую визуализацию и учёт привычек, но не позволяет описать отдельное событие и реакцию на него в терминах КПТ; набор эмоций ограничен, автоматический анализ текста нейросетью не предусмотрен.

Сравнение показало, что у популярных решений в целом хорошо реализованы фиксация настроения и визуализация, однако полный цикл СМЭР, шкала интенсивности, справочник эмоций по Плутчику и нейросетевой разбор текста представлены не в полном объёме. Это обосновало необходимость разработки собственного приложения «Дневник эмоций».

После анализа аналогов сформулирован набор ключевых функций разрабатываемого приложения:

1. ведение записей по протоколу СМЭР с четырьмя обязательными текстовыми полями и настраиваемой датой события;
2. ручной выбор эмоций из справочника Плутчика (32 состояния) с интенсивностью от 1 до 10;
3. подбор эмоций по тексту полей СМЭР с помощью локальной нейросети;
4. просмотр, редактирование и удаление накопленных записей с фильтрацией по периоду;
5. отображение статистики распределения эмоций за выбранный интервал;
6. опциональный сводный текстовый анализ нескольких записей через GigaChat после согласия пользователя.

Для реализации выбран технологический стек: мобильный клиент на React Native (Expo), серверная часть на Node.js (Express), СУБД PostgreSQL, отдельный ML-сервис на Python (Flask) и опциональный сводный анализ через API GigaChat. React Native позволяет описывать интерфейс декларативно и переиспользовать логику на iOS и Android; PostgreSQL обеспечивает реляционное хранение связей между пользователями, записями и эмоциями; выделение ML-сервиса в отдельный модуль упрощает обновление модели без перевыпуска мобильного приложения. Такая архитектура обеспечивает модульность, независимое развитие клиента, сервера и сервиса классификации, а также разделение локальной обработки текста при подборе эмоций и облачного анализа по согласию пользователя.

Выводы по первому разделу: проведённый обзор подтвердил актуальность темы, позволил сформулировать функциональные требования к приложению «Дневник эмоций» и определить технологический стек программного комплекса.

Второй раздел «Разработка и реализация мобильного приложения» посвящён проектированию, реализации и проверке программного продукта.

Основной упор сделан на самостоятельную разработку программного комплекса: мобильного клиента, серверной части, базы данных, ML-сервиса и интеграции с GigaChat.

Определена целевая аудитория: студенты и молодые специалисты при высокой учебной и рабочей нагрузке, клиенты психологов когнитивно-поведенческого направления, ведущие дневник между сессиями как домашнее задание, а также пользователи с повышенной тревожностью, которым нужна понятная структура для описания ситуации, мыслей и реакций.

Три характерных сценария иллюстрируют связь шагов работы с функциями приложения. Сценарий 1 (ежедневная запись): пользователь фиксирует произошедший случай, заполняет ситуацию и автоматические мысли, нажимает «Подобрать эмоции (нейросеть)», получает предложенные состояния с интенсивностью, при необходимости корректирует список, дописывает телесные ощущения и поведение; запись сохраняется и отображается в списке за выбранный период. Сценарий 2 (ретроспектива): пользователь вносит или правит запись за прошедшую дату; при сохранении передаётся актуальная версия записи, а при расхождении версий сервер возвращает отказ в обновлении, чтобы не перезаписать более новые данные. Сценарий 3 (рефлексия за период): на экране статистики пользователь выбирает месяц, видит преобладание отдельных эмоций, запускает анализ через GigaChat и читает текст с паттернами и рекомендациями для самонаблюдения.

Сформулированы требования к системе: регистрация и вход с выдачей JWT-токена; создание записи СМЭР с четырьмя обязательными текстовыми полями и настраиваемой датой; привязка не менее одной эмоции из справочника Плутчика с интенсивностью 1–10; подбор эмоций по кнопке на форме (не более пяти предложений); статистика распределения эмоций за выбранный период; сводный текстовый отчёт через GigaChat после явного согласия пользователя. Среди нефункциональных требований — хеширование паролей, изоляция дан-

ных по пользователю, оптимистичная блокировка через поле `version`, описание REST API в формате OpenAPI со Swagger UI, развёртывание PostgreSQL через Docker Compose и приемлемое время отклика API при локальном тестировании.

Программный комплекс построен по клиент-серверной архитектуре (рисунок 1). Мобильное приложение (React Native, Expo) отвечает за интерфейс, навигацию и HTTP-запросы; сервер Node.js — за REST API, аутентификацию и бизнес-логику; PostgreSQL — за хранение пользователей, записей СМЭР, справочника эмоций и связей `entry_emotions`. Подбор эмоций по тексту выполняет отдельный ML-сервис на Python (Flask); сводный анализ при необходимости запрашивается через API GigaChat. Мобильный клиент не обращается к базе данных и ML-сервису напрямую.

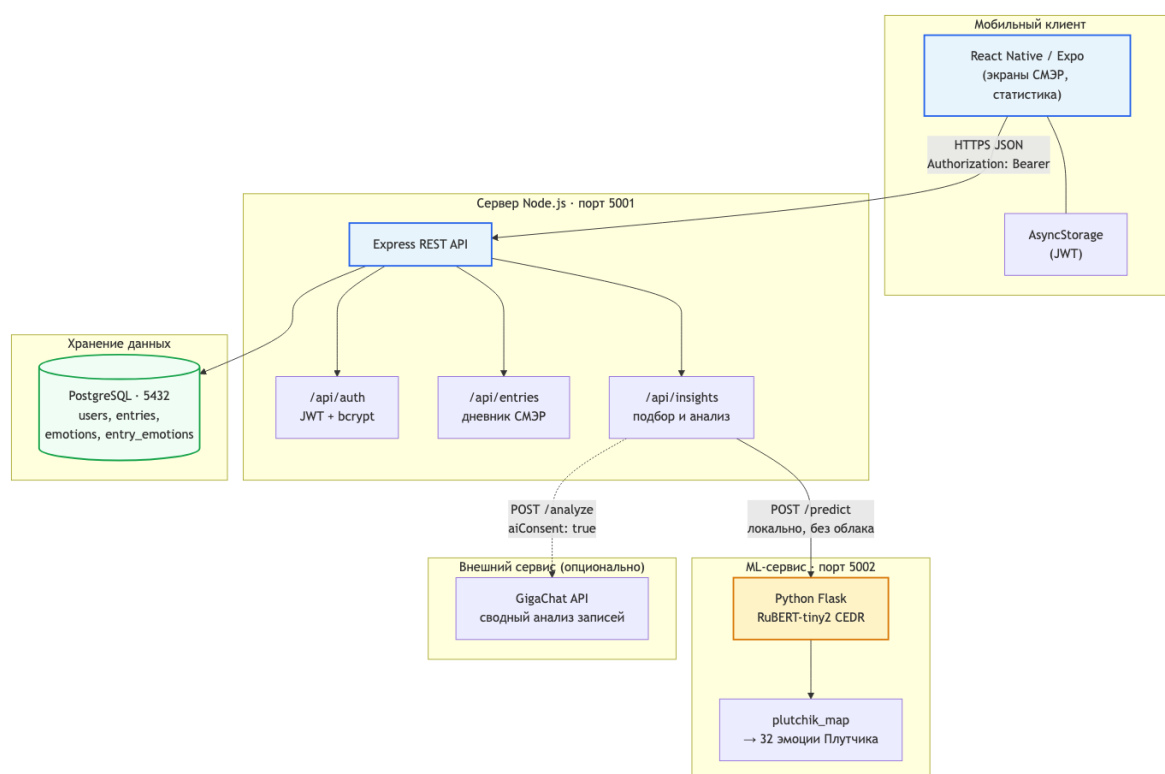


Рисунок 1 – Диаграмма архитектуры программного комплекса «Дневник эмоций»

Модель данных приведена к третьей нормальной форме: таблицы `users`, `entries`, `emotions`, `entry_emotions`. Поле `version` используется для обнаружения конфликта при одновременном редактировании записи (код ответа 409). При выдаче списка записей сервер формирует JSON с вложенным массивом эмоций, чтобы клиент одним запросом получил данные для списка, фильтра по периоду и экрана статистики.

REST API — единственный канал обмена между мобильным клиентом и

сервером. Для чтения используется GET, для создания — POST, для изменения — PUT или PATCH, для удаления — DELETE. Тела запросов и ответов — JSON в кодировке UTF-8. Контракт зафиксирован в спецификации OpenAPI; для ручной проверки доступны Swagger UI и Postman. Ключевые маршруты включают: /api/auth/register и /api/auth/login — регистрация и вход; /api/entries — список и создание записей; /api/entries/:id — просмотр, правка и удаление одной записи; /api/emotions-list — справочник эмоций; /api/emotions — привязка эмоции к записи; /api/insights/suggest-emotions — подбор эмоций по тексту СМЭР; /api/insights/analyze — сводный анализ через GigaChat. Мобильный клиент не обращается к базе данных и ML-сервису напрямую: операции дневника и подбор эмоций проходят через REST API на Node.js, который при необходимости обращается к локальному ML-сервису на Python.

Реализована подсистема аутентификации. На экранах регистрации и входа пользователь вводит email, имя, фамилию и пароль; при успешном ответе сервер возвращает JWT-токен, сохраняемый на устройстве. Пароль хранится на сервере только в виде хеша (bcrypt); защищённые маршруты проверяются middleware, извлекающим и верифицирующим токен из заголовка Authorization: Bearer. Запросы к записям и аналитике всегда фильтруются по идентификатору пользователя из токена. Сессия восстанавливается после перезапуска приложения по сохранённому токену и принудительно завершается при выходе или ошибке авторизации.

Реализована подсистема ведения дневника СМЭР. Форма записи содержит четыре многострочных поля — ситуация, мысли, телесная и поведенческая реакция — и блок работы с эмоциями. Дату события можно задать через datePicker, в том числе задним числом. Справочник из 32 эмоций загружается при открытии формы; на запись допускается несколько эмоций с интенсивностью, задаваемой ползунком 1–10. Сохранение выполняется в два шага: создание записи и привязка эмоций. Главный экран списка отображает карточки записей с датой и началом текста ситуации; нажатие открывает детальный просмотр с полным текстом СМЭР и чипами эмоций. Фильтрация по периоду реализована через модальное окно PeriodPicker; значения сохраняются в PeriodPickerContext и применяются и к списку, и к экрану статистики. На экране статистики вычисляются частота эмоций и средняя интенсивность за выбранный интервал.

Реализована подсистема анализа текста. Она включает два компонента: локальный подбор эмоций по полям СМЭР и опциональный сводный отчёт по накопленным записям.

Для подбора эмоций выбран компактный русскоязычный трансформер RuBERT-tiny2 (≈ 29 млн параметров), изначально обученный на корпусе CEDR. Корпус CEDR (sagteam/cedr_v1) содержит 9410 русскоязычных предложений из Twitter, LiveJournal и Lenta.ru с мультиметочной разметкой по пяти базовым эмоциям; обучающая выборка — 7528 примеров, тестовая — 1882. В проекте сеть дополнительно дообучена под формат полей дневника СМЭР. Подготовка и применение модели включает пять этапов:

1. согласование меток датасета CEDR и шестиклассного выхода модели (модуль `cedr_labels.py`);
2. аугментация части обучающих примеров в формат СМЭР (доля — 0,5);
3. дообучение классификатора скриптом `train_cedr.py` (3 эпохи, скорость обучения 10^{-5} , длина контекста 256 токенов);
4. применение модели в ML-сервисе `server.py` с отбором меток по порогам уверенности;
5. сопоставление базовых классов CEDR с 32 состояниями Плутчика модулем `plutchik_map.py`.

Задача формулируется как мультиметочная классификация шести классов (отсутствие эмоции, радость, грусть, удивление, страх, злость). На тестовой выборке CEDR после дообучения достигнуты $F1$ (микроусреднение) = 0,785, $F1$ (макроусреднение) = 0,719 и функция потерь на тесте = 0,246. При работе сервиса вероятности по классам вычисляются через сигмоиду; отбираются метки с уверенностью не ниже порога 0,18, при необходимости список дополняется наиболее вероятными классами при уверенности не ниже 0,06. Базовые метки CEDR сопоставляются с тремя градациями интенсивности в справочнике Плутчика (например, для «грусть» — «Разочарование», «Грусть», «Горе»); интенсивность для шкалы 1–10 вычисляется как округлённое произведение уверенности модели на 10.

После заполнения полей «Ситуация» и «Мысли» пользователь нажимает «Подобрать эмоции (нейросеть)»; сервер собирает текст, обращается к локальному ML-сервису и возвращает список предложений с интенсивностью и текстом `disclaimer`. Пользователь может скорректировать список перед сохране-

нием. Подбор эмоций выполняется локально — без передачи текста во внешний коммерческий сервис для этой функции. Условный пример: для текста о дожде и одиночестве модель с высокой вероятностью выделяет класс «грусть» и предлагает эмоцию «Грусть» с интенсивностью 7; финальное решение остаётся за пользователем.

Сводный анализ накопленных записей реализован через GigaChat. Пользователь нажимает кнопку анализа на экране статистики; открывается диалог согласия AiConsentModal. Сервер выбирает до 30 последних записей пользователя, сериализует их в единый текст и формирует запрос с системной инструкцией: роль помощника по рефлексии, структура ответа — «Анализ паттернов», «Тенденции», «Рекомендации», запрет медицинских диагнозов. Готовый текст возвращается в поле analysis и не сохраняется в базе. При недоступности GigaChat (код 503) дневник и локальный подбор эмоций продолжают работать. Разработанное приложение позиционируется как инструмент самонаблюдения и учебной апробации, а не медицинское изделие; подсказки нейросети носят ориентировочный характер.

Проведено функциональное и нефункциональное тестирование. Тестовая среда включала клиента на iOS-эмуляторе и физическом Android-устройстве, сервер Node.js, PostgreSQL (Docker Compose) и Python-сервис классификации. На уровне REST API проверены успешные операции и негативные сценарии: неверный пароль (401), создание записи без токена (401), правка записи с устаревшей version (409), сводный анализ без ключей GigaChat (503), подбор эмоций при остановленном ML-сервисе (503). В мобильном клиенте проверено выполнение требований:

1. регистрация, вход, выдача JWT, сохранение сессии после перезапуска, выход;
2. запись СМЭР: обязательные текстовые поля, настраиваемая дата, выбор эмоции из справочника;
3. не менее одной эмоции на запись, интенсивность 1–10;
4. подбор эмоций по кнопке (не более пяти, с интенсивностью);
5. распределение эмоций за выбранный период;
6. сводный текстовый отчёт через GigaChat после подтверждения согласия.

По всем перечисленным сценариям зафиксирован успешный результат. Дополнительно подтверждена изоляция данных (записи видны только владельцу JWT),

устойчивость к отказу GigaChat и ML-сервиса (дневник и ручной выбор эмоций сохраняются), приемлемое время загрузки списка записей при десятках карточек и корректная реакция на типовые ошибки ввода. Полный пользовательский цикл — от регистрации до сводного анализа — проходит за несколько минут без обращения к вспомогательным инструментам диагностики API.

Выводы по второму разделу: разработан и протестирован программный комплекс «Дневник эмоций», реализующий дневник СМЭР с локальной нейросетевой классификацией эмоций и опциональным сводным анализом записей. Практическая часть подтверждена функциональным и нефункциональным тестированием основных и негативных сценариев.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы был проведён обзор психологических и технологических подходов к цифровой саморегуляции эмоциональных состояний. Рассмотрены модель ABC, дневник СМЭР, современные методы анализа русскоязычных текстов и существующие программные аналоги. Это позволило обосновать архитектурные и технологические решения проекта и сформулировать функциональные требования к мобильному приложению.

В результате спроектировано и разработано мобильное приложение для саморегуляции эмоциональных состояний на основе модели ABC и дневника СМЭР, интегрированного с нейронной сетью для автоматической классификации эмоционального содержания текста пользователя. Программный комплекс включает мобильный клиент на React Native, серверную часть на Node.js и PostgreSQL, отдельный ML-сервис на Python. Реализованы ключевые функции: регистрация и аутентификация пользователя, создание, редактирование и удаление записей СМЭР, привязка эмоций с интенсивностью, фильтрация по периоду и визуализация статистики.

Дополнительно реализованы функции обработки текста с помощью средств NLP-анализа нейронных сетей: локальное определение эмоциональной окраски и подбор эмоций без передачи данных во внешние сервисы и опциональный сводный анализ через GigaChat при явном согласии пользователя. Такое разделение позволяет сочетать удобство повседневного ввода с расширенной аналитикой и контролем приватности.

Проведённое тестирование подтвердило корректную работу основных и негативных сценариев, устойчивость к типовым ошибкам ввода и сетевым сбоям, а также согласованность данных между мобильным клиентом и сервером. Результаты показывают, что поставленная цель достигнута, а сформулированные задачи выполнены. Разработанное решение может использоваться как инструмент структурированного самонаблюдения и служить основой для дальнейшего развития цифровых сервисов поддержки эмоционального благополучия.