

Ковалев Глеб 367291				
Задание №2. Идентификация информационной системы в соответствии с названием и кратким описанием		7. HealthNavigator Описание: Приложение для мониторинга здоровья, использующее носимые устройства для отслеживания жизненных показателей и предлагающее рекомендации по улучшению самочувствия.		
№	Классы системы	Краткое обоснование принадлежности		
1	Низовой уровень	Система помогает мониторить здоровье отдельным людям		
2	Пользовательская ИС	Пользователь напрямую взаимодействует через мобильное приложение		
3	Частично интегрированная	Интеграция с системами носимых устройств, при согласии - обмен с внешними системами (медкарты), ядро автономное		
4	Облачное	Хранение и обработка агрегированных данных, обучение моделей рекомендаций, глобальная масштабируемость		
	Мобильное	Основной клиент - смартфон или часы. Собираются данные с датчиков, пользователь локально уведомляется		
	Веб-ориентированное	Веб-панель для основных данных о здоровье, которые можно прислать врачу		
	Клиент-серверное	Разграничиваем ответственности клиента и сервера для реализации мобильного и веб приложений		
5	Графический интерфейс	Для понятной визуализации данных		
6	Обработка текстовых данных	Подразумеваются данные системного уровня. Как то: числа, состояния (текст), динамика изменения (графики)		
	Обработка числовых данных	Обрабатываем данные, которые мы получаем с устройств		
7	Управленческая ИС	Назначение ИС - поддерживать принятие решений пользователем (или врачом). Оно анализирует здоровье и помогает пользователю улучшить здоровье		
№	Архитектура	Плюсы и минусы		
1	Монолитная	Быстрая реализация, проще тестирование и поиск ошибок, но сложно масштабировать, длинные релизы и интеграция нескольких устройств будет приводить к излишнему усложнению кода		
2	Клиент-серверная	Естественная модель для мобильных приложений, централизованная безопасность, удобно делать резервное копирование данных, но есть опасность с нагрузкой на сервер, если будут потоки данных сервер может не справиться		
3	SOA	Удобное разделение задач, можно использовать разные технологии для разных задач, легко подключать внешние системы, но сложно настроить взаимодействие сервисов, слишком громоздко для небольших команд		
4	Микросервисная	Можно масштабировать только нужную часть, сбой в одном сервисе не будет ломать приложение, но сложнее управлять и обновлять сервисы, нужно много инструментов DevOps, сложно поддерживать консистентность данных		
5	Компонентноориентированная архитектура	Чистый код, легкое внедрение новых устройств, удобно дорабатывать или отключать части, но если что-то упадет - может упасть все и необходимо следить за границами между компонентами		
6	Событийно-ориентированная	Идельная архитектура для данных с устройств, слабая связанность, проще обрабатывать потоки телеметрии, но трудно понять, что происходит, нужно следить, чтоб события не терялись, нужны брокеры сообщений		
7	Облачная	Легкая масштабируемость, нет необходимости покупать свои сервера, есть доступ к мощным инструментам, но может быть дорого при большом объеме данных, мы зависимы от интернета и важно соблюдать законы по защите персональных данных (особенно медицинских)		