

Адаптивное мобильное приложение для психодиагностики и профориентации на основе графа тестов

Носаченко А.А. (ИГУ)

**Научный руководитель – кандидат физико-математических наук, доцент
Пещерова С.М. (ИГУ)**

Введение. Традиционные методы профориентации и психодиагностики часто представляют собой набор линейных тестов. Пользователь вынужден проходить фиксированное количество вопросов, даже если промежуточные ответы уже четко определили его профиль. Это приводит к утомляемости, снижению правдивости и увеличению времени диагностики. При помощи реализации адаптивной системы подбора вопросов можно решить эти проблемы [1].

Основная часть. Основой программного комплекса для реализации адаптивного тестирования является графовая модель, которая обеспечивает персонализированный маршрут тестирования. Сервис реализован в виде мобильного приложения, что позволяет в комфортном темпе проходить тестирование и отслеживать результаты тестов поэтапно, также обеспечивается высокая доступность для пользователя и отсутствие необходимости в установке дополнительного ПО.

Система динамически строит траекторию опроса: вместо линейного прохождения фиксированного списка вопросов алгоритм анализирует ответы пользователя в реальном времени и подбирает следующий блок вопросов. Это позволяет исключить ненужные вопросы и углубиться в значимые характеристики опрашиваемого.

Особое внимание стоит уделить программному обеспечению, разработкой которого занимался автор. Оно состоит из модулей, которые описаны ниже.:

1. База данных, развернутая на платформе Supabase (PostgreSQL). Здесь реализована графовая структура. Таблица `test_nodes` хранит узлы (как вопросы, так и информационные блоки), а таблица `test_edges` описывает связи между ними. Ключевой особенностью является использование условных переходов, зависящих от накопленных баллов в шкалах, а не от последнего ответа.

2. Логика графа, написанная на языке Python. Данный модуль отвечает за навигацию по графу. При достижении порогового значения суммы ответов от пользователя, система переходит к следующему подграфу. Это обеспечивает адаптацию сценария опроса под историю ответов респондента.

3. Модуль интерфейса. Он преобразует данные узла в визуальный формат и обеспечивает взаимодействие с пользователем. Выбор Telegram обусловлен возможностью сосредоточиться только на реализации логики движка графов, без дополнительных время- и трудозатрат на фронтенд-разработку.

Выводы. Разработанная архитектура позволяет объединить отдельные методики тестирования в единый граф. При этом, способ взаимодействия остается интуитивно понятным, что снижает когнитивную нагрузку на пользователя и повышает проходимость тестов до конца, по сравнению с бумажными или веб-формами, кроме этого, значительно количество вопросов и длительность самого тестирования.

Данное решение позволяет гибко настраивать логику переходов: добавлять методики или изменять веса вопросов, путем редактирования базы данных, без переписывания программного кода. С учетом накопления данных о прохождении тестов, в будущем

планируется добавление алгоритма SPMiner для автоматического выявления наиболее эффективных последовательностей вопросов, отталкиваясь от параметров предыдущего подграфа [2].

Список использованных источников:

1. Liu Q. et al. Survey of Computerized Adaptive Testing: A Machine Learning Perspective // arXiv preprint arXiv:2404.00712. – 2024. – DOI: 10.48550/arXiv.2404.00712.
2. Ying R. et al. Representation Learning for Frequent Subgraph Mining // arXiv preprint arXiv:2402.14367. – 2024. – DOI: 10.48550/arXiv.2402.14367.