

ритма, тем он лучше; чем меньше внутрикластерное расстояние, тем больше точки сконцентрированы внутри кластера, тем эффективнее метод; чем больше межкластерное расстояние, тем дальше кластеры удалены друг от друга, тем эффективнее алгоритм.

Результаты классификации пациентов на основе анализа данных об экспрессии генов представлены для шести методов снижения размерности данных на рисунке. Наилучшие результаты показал алгоритм UMAP, отличительной особенностью которого является построение взвешенного неориентированного графа более низкой размерности. Алгоритм показал наименьшее значение $\langle d_{intra} \rangle = 50,41$, наибольшее значение $\langle d_{inter} \rangle = 2,28$, и наименьшее отношение $\frac{\langle d_{intra} \rangle}{\langle d_{inter} \rangle} = 22,15$.

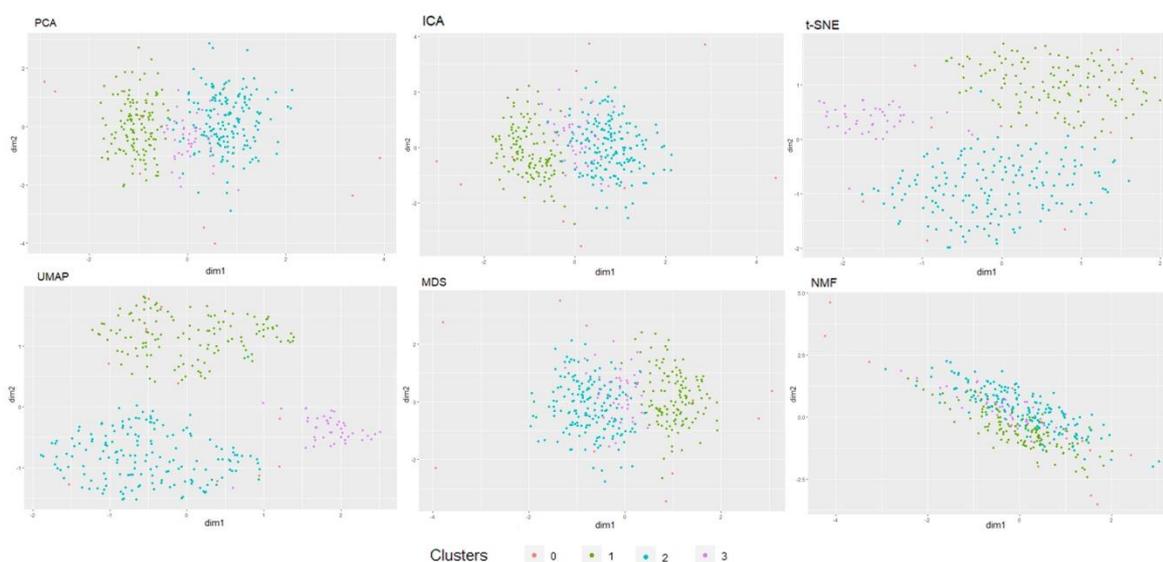


Рис. 1. Диаграммы разброса для кластеров данных системы 3 в пространствах наиболее информативных координат, вычисленных методами PCA, ICA, tSNE, UMAP, MDS, NMF

Методы tSNE и UMAP даже на сложных данных успешно выполняют задачу, однако UMAP работает качественнее, в tSNE наблюдается менее выраженное разделение, в то время как после выполнения остальных методов наблюдается частичное или полное перекрытие нескольких кластеров. По степени разделения данных наилучшие результаты демонстрирует метод UMAP. Значения $\langle d_{intra} \rangle$ и $\langle d_{inter} \rangle$ наихудшие у NMF. PCA имеет низкие показатели по рассматриваемым критериям, однако обладает более высокой скоростью работы. Самый быстрый алгоритм ICA плохо справляется с зашумленными данными, наблюдается значительное перекрытие кластеров. MDS показывает схожие результаты с ICA, однако проигрывает по времени работы. Оценки критериев качества работы алгоритмов представлены в таблице.

**Оценки критериев качества работы алгоритмов
на смоделированных наборах данных**

Система	Параметр	Алгоритм					
		PCA	ICA	tSNE	UMAP	MDS	NMF
Система 1	t, c	0,02	0,01	15,42	1,38	0,22	0,22
	$\frac{\langle d_{intra} \rangle}{\langle d_{inter} \rangle}$	28,99	13,27	11,74	5,19	13,27	10,70
Система 2	t, c	0,01	0,01	15,11	1,41	0,27	0,23
	$\frac{\langle d_{intra} \rangle}{\langle d_{inter} \rangle}$	51,18	28,10	15,18	10,05	28,10	19,58
Система 3	t, c	0,03	0,02	7,57	1,06	0,11	0,25
	$\frac{\langle d_{intra} \rangle}{\langle d_{inter} \rangle}$	50,19	54,15	26,20	22,15	54,15	112,95

ЗАКЛЮЧЕНИЕ

Выполнен сравнительный анализ шести методов снижения размерности данных: PCA, ICA, tSNE, UMAP, MDS, NMF. Лучше всего с задачей разделения кластеров данных об экспрессии генов справляется UMAP (среднее отношение кластерных расстояний 12,5). Методы PCA (43,5) и NMF (47,7) подходят преимущественно для чистых данных невысокой размерности, при усложнении данных наблюдается большое перекрытие кластеров. Самый быстрый из представленных метод ICA (31,8) и сходный в работе с ним метод MDS (31,8) следует использовать на чистых или средних данных. Метод tSNE (17,7) отлично подходит для разделения чистых и средних данных, на зашумленных данных метод работает приемлемо, но хуже, чем UMAP. При увеличении начальной или конечной размерности данных, tSNE значительно проигрывает UMAP.

Выполненная работа позволяет сделать вывод о том, что наиболее оптимальными алгоритмами снижения размерности данных для исследования экспрессии генов являются методы tSNE и UMAP.

Библиографические ссылки

1. Li, X. Genomic Analysis of Liver Cancer Unveils Novel Driver Genes and Distinct Prognostic Features / X. Li, W. Xu, W. Kang // *Theranostics*. 2018. № 8. P. 1740–1751.
2. Maaten, L., Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research*. 2008. Vol. 9, P. 2579-2605.
3. Qiu W., Joe H. Generation of Random Clusters with Specified Degree of Separation // *Journal of Classification*. 2006. Vol. 23. DOI: 10.1007/s00357-006-0018-y.

АВТОМАТИЗАЦИЯ РАБОТЫ СОТРУДНИКОВ КУХНИ ОБЪЕКТОВ ОБЩЕСТВЕННОГО ПИТАНИЯ

И. Д. Горшков

Белорусский государственный университет, г. Минск;

mailthedump@gmail.com;

науч. рук – А. В. Микулович, ст. преп.

В работе рассматриваются вопросы, связанные с разработкой мобильного приложения, предназначенного для структурирования и частичной автоматизации работы сотрудников объектов общественного питания. Реализация мобильного приложения производилась с помощью интегрированной среды разработки Xcode, предоставляющей весь необходимый функционал по созданию программно-аппаратной и графической части приложения. Интегрированная среда разработки написана на языке программирования Objective-C. Ее компоненты позволяют создавать программный код, графическую часть приложения и тестировать все возможные части кода с интерфейсом приложения. Логика приложения, реализуемая программным кодом и графическим интерфейсом, вынесены в отдельные файлы, что позволяет обеспечить полное покрытие тестами программного кода. Подобный подход позволяет грамотно использовать все компоненты интегрированной среды разработки Xcode, что в конечном итоге приводит к стабильной и отлаженной работе мобильного приложения.

Ключевые слова: Интегрированная среда разработки Xcode; мобильная разработка; objective-c; графический интерфейс; реализация мобильного приложения.

ВВЕДЕНИЕ

В условиях увеличения населения и популяризации общепита грамотное принятие и обработка заказов становятся особенно важными. По результатам опроса, проведенного среди работников трех ресторанов (IL Патио, Union Coffee, Cafe Netto), все были согласны, что с помощью специализированного мобильного приложения они бы ускорили свою работу. Все три ресторана работали по одному методу обработки заказа: первым шагом было принятие заказа у клиента, вторым шагом – оформление заказа через специализированный терминал и третий шаг – появление чека на кухне для начала подготовки блюда.

В настоящее время на рынке не существует мобильных приложений, которые могли бы в полной мере автоматизировать работу с помощью мобильных устройств. Из-за этого официантам приходится тратить большое количество времени, чтобы оформить заказ и передать его на кухню, что в итоге приводит к убыткам ресторанов. В связи с этим было принято решение о разработке мобильного приложения, которое позволит официантам намного быстрее принимать заказы, оформлять их, а также отслеживать, какие столы они обслуживали в свою рабочую смену.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ К ПРИЛОЖЕНИЮ

Перед началом создания мобильного приложения с помощью интегрированной среды разработки Xcode [1] необходимо разработать техническое задание [2] и прототипы экранов приложения [3]. Техническое задание позволяет четко понимать, какой функционал должен существовать в будущем приложении и каковы примерные сроки разработки. На основе прототипов экранов (рис. 1) и задач, описанных в техническом задании, строится пошаговый план создания мобильного приложения.

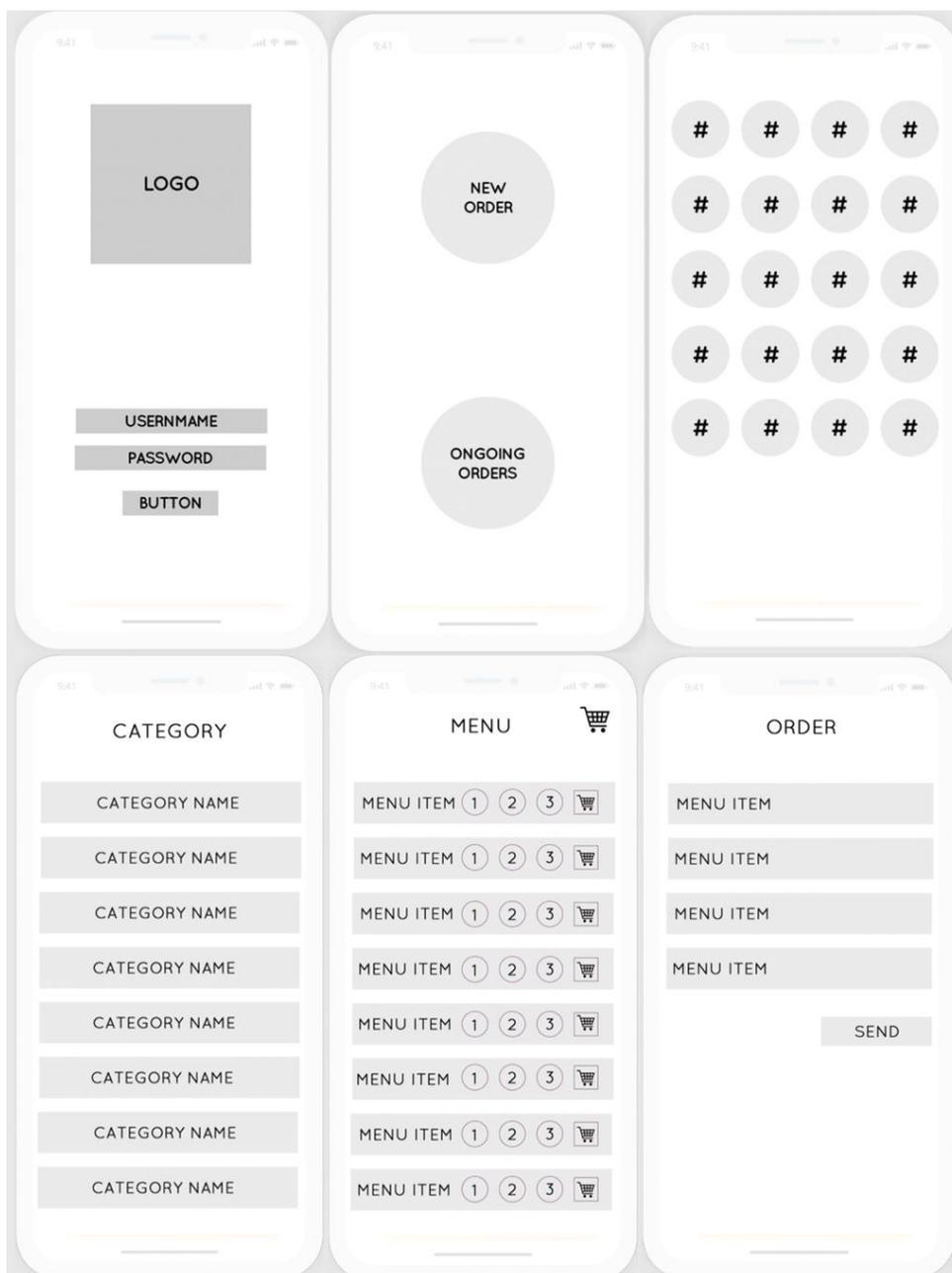


Рис. 1. Интерфейс прототипа приложения

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ

Согласно техническому заданию приложение состоит из 6 экранов и имеет следующий функционал (рис. 2):

1. Логирование пользователя.
2. Создание нового заказа или просмотр текущих.
3. Выбор стола.
4. Выбор категории блюда.
5. Выбор блюда.
6. Проверка заказа и завершение формирования заказа.

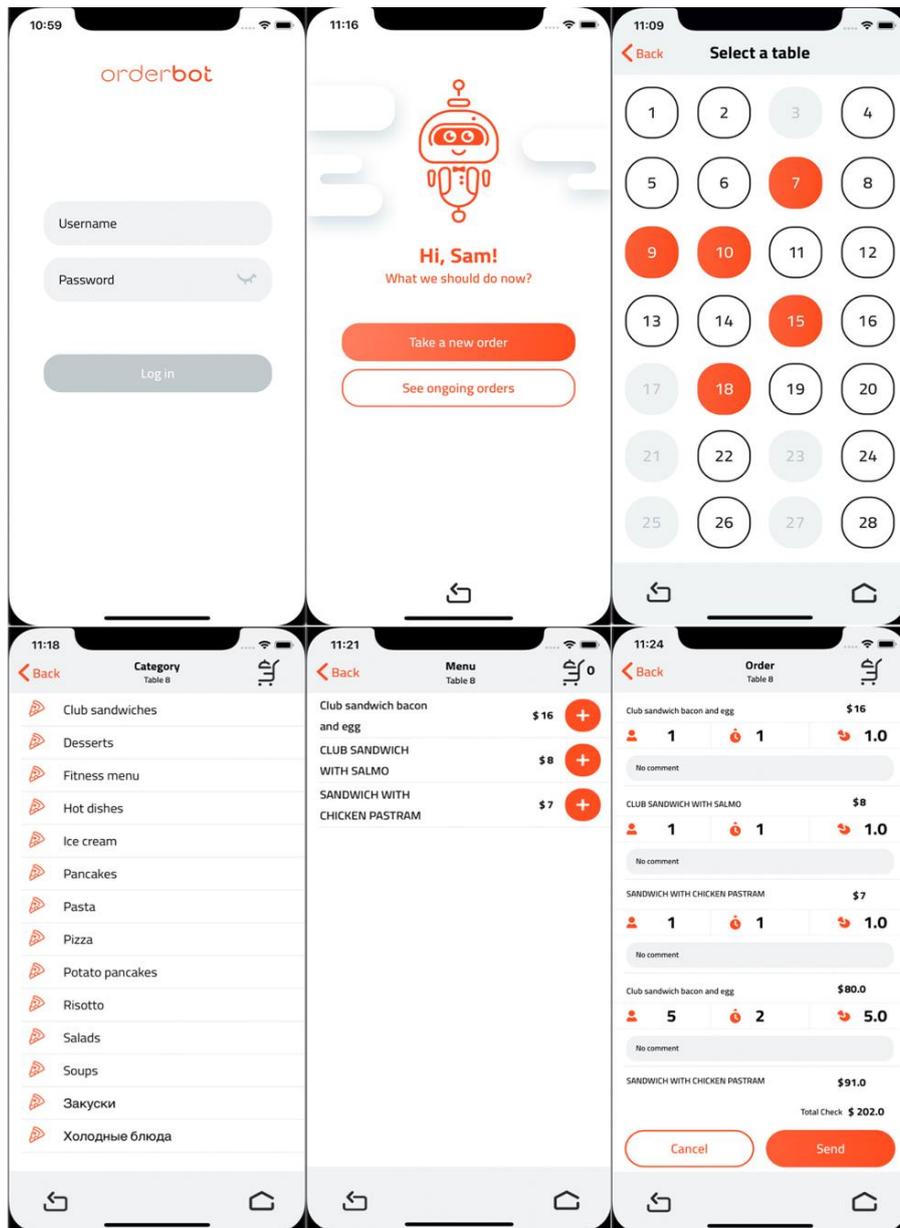


Рис. 2. Интерфейс разработанного приложения

В ходе работы было разработано мобильное приложение для автоматизации работы сотрудников кухни объектов общественного питания. В приложении был учтен весь описанный функционал технического задания. Графический интерфейс разработан в соответствии с созданными прототипами. Для разработки использовалась интегрированная среда разработки Xcode.

Была реализована основная идея – создать простое приложение, которое помогало бы легко и просто оформить заказ. При этом был учтён ряд недостатков уже существующих методов принятия заказов: долгое оформление заказа, лишняя посещаемость терминала, исправление ошибок официанта только через терминал, невозможность узнать готовность заказа без прихода на кухню. Интерфейс настоящего приложения достаточно прост и не требует долгого изучения.

ВЫВОДЫ

Данное приложение может упростить и ускорить работу сотрудников кухни объектов общественного питания. Простой графический интерфейс позволяет быстро оформлять, редактировать и отслеживать заказы. За счет реализации приложения через интегрированную систему Xcode в будущем будет довольно просто добавлять новый функционал и вносить изменения в существующий графический интерфейс. На данный момент приложение поддерживает все существующие версии мобильных телефонов iPhone и операционные системы iOS.

Библиографические ссылки

1. Xcode [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/Xcode> (дата обращения: 21.05.2020).
2. Техническое задание на разработку приложения [Электронный ресурс]. URL: <https://punicapp.com/blog/pages/332/tehnicheskoe-zadanie-na-razrabotku-prilozheniya/> (дата обращения: 21.05.2020).
3. Прототипирование приложения [Электронный ресурс]. URL: <http://improve-group.ru/materials/prototyping-review.html> (дата обращения: 21.05.2020).