

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра системного анализа и автоматического управления

**РАЗРАБОТКА С ПРИМЕНЕНИЕМ UML СИСТЕМЫ
АВТОМАТИЗИРОВАННОЙ ДОСТАВКИ ПРОДУКЦИИ КАФЕ**
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 5 курса 551 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Никонорова Дмитрия Сергеевича

Научный руководитель

к. т. н., доцент

Д. Ю. Петров

Заведующий кафедрой

к. ф.-м. н., доцент

И. Е. Тананко

Саратов 2022

ВВЕДЕНИЕ

Актуальность темы. В условиях современного рынка постоянно усиливается конкуренция, а также рост издержек. Поэтому для благополучного функционирования любой организации, в том числе и оказывающей услуги в сфере общественного питания, необходима результативная работа с клиентами. Эту работу, как правило, выполняет отдел по работе с клиентами или отдел продаж, которые занимаются оформлением документов, ведением дел, а также организацией учета клиентов. Все эти функции являются достаточно трудоемкими, что снижает эффективность каждого работника. Сбор исходной информации является особенно трудоемкой задачей, которая негативно отражается на скорости и производительности специалистов отдела, что в свою очередь затрудняет выполнение целей поставленных руководством. В тоже время своевременное получение подлинной и точной информации способствует удержанию клиентов специалистом. Отсюда следует необходимость в определенной степени автоматизации некоторых видов деятельности таких отделов работающих с клиентами [1, 2].

Ввиду того, что в России, как и во многих других странах, бизнес в сфере общественного питания представляет собой одной из наиболее динамично прогрессирующих отраслей частного предпринимательства, спрос на автоматизированные системы для соответствующих предприятий стабильно растет. Автоматизация становится неким стандартом, который необходим для достижения конкурентоспособности этого бизнеса. За последние годы появилось огромное количество молодых компаний, которые специализируются на автоматизации услуг в этой сфере. Помимо этого, в развитие данной области вносят вклад, также и некоторые именитые бренды, например 1С.

Актуальность автоматизации процесса доставки реализуемой продукции кафе клиенту заключается в увеличении рентабельности, снижении издержек организации, а также в повышении производительности труда рабочего персонала, в наиболее надежном, удобном и быстром методе предоставления данных о стадиях выполнения и степени завершенности заказа, а так же в скорейшем выполнении заказа.

Цель бакалаврской работы — описание необходимой для ведения бизнеса системы автоматизированной доставки продукции кафе с применением на практике унифицированного языка моделирования UML (Unified Modeling

Language). В данном случае использование этого языка должно облегчить взаимодействие исполнителя с заказчиком, а также внутри главной команды исполнителей и/или сторонними командами исполнителей, в случае перераспределении объемов выполняемых работ. Также использование UML должно облегчить дальнейший процесс написания исполняемого программного кода за счет возможности кодогенерации для выбранного языка программирования и обеспечить простоту релокации частей разрабатываемой системы на другие платформы. Таким образом в целом назначение этого комплекса состоит в визуальном представлении структуры и процесса работы заданной системы для детальной проработки на этапе анализа, а также упрощения ее последующей реализации.

Поставленная цель определила **следующие задачи:**

1. проведение анализа требований;
2. проектирование комплекса необходимых UML–моделей для дальнейшей разработки системы;
3. программная реализация алгоритмов той части системы, которая отвечает за взаимодействие сервера с клиентом и базой данных, для оценки состоятельности спроектированного комплекса UML–моделей. Эта часть выбрана ввиду особой важности, заключающейся в наибольшей ориентированности на потенциального клиента, что само по себе является отправной точкой при реализации системы в целом.

Практическая значимость бакалаврской работы. Все результаты, полученные при решении задач, могут быть использованы для дальнейшего развития системы и ее внедрения на предприятиях оказывающих услуги в сфере общественного питания.

Структура и объем работы. Бакалаврская работа состоит из введения, двух разделов, заключения, списка использованных источников и приложений. Общий объем работы — 117 страниц, из них 63 страницы — основное содержание, включая 27 рисунков, 24 приложения, список использованных источников информации — 34 наименования.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Анализ систем автоматизированной доставки еды из кафе и теоретические основы унифицированного языка UML» посвящен исследованию особенностей процесса заказа еды клиентом и ее доставки, систем, которые реализуют такой процесс, а также общему обзору самого языка UML вместе с наиболее часто используемыми при моделировании диаграммами и принципами их построения.

Сфера общественного питания на сегодняшний день является одной из самых динамично эволюционирующих отраслей. По оценкам разных специалистов, среднегодовые темпы роста бизнеса в этой сфере на российском рынке составляют ориентировочно 25–30%, и эта динамика будет оставаться такой еще, как минимум, на протяжении ближайших трех лет. Специалисты объясняют это, в первую очередь, тем, что условия жизни основной части населения постоянно улучшаются и увеличивается ее доходность, а это также способствует росту числа потенциальных клиентов кафе и развлекательных центров [1, 2].

На текущий момент, как в России, так и в остальных странах численность подвергнутых автоматизации ресторанов, кафе, фаст-фудов, а также корпоративных предприятий питания, в том числе, и принадлежащим им сетей, насчитывает десятки тысяч, и при этом продолжает стремительно расти [1].

Агрегаторами называют электронные торговые площадки, которые обеспечивают осуществление закупок «в один приём». Таким образом агрегатор выполняет рационализированное снабжение за счет использования множества каталогов, которые ориентированы по группам покупателей. Модель агрегатора выполняет поддержку процесса закупок вплоть до заключения контрактов по поставке с различными продавцами, а также снабжает покупателя информацией о текущем статусе поставки. Здесь отдельным подвидом выступают агрегаторы товаров, которые дают пользователям возможность выбирать и сравнения товаров сразу из множества магазинов.

На российском рынке действуют два крупнейших и пожалуй наиболее известных сервиса, которые специализируются на заказах, а также доставке еды на дом, это Яндекс.Еда и Delivery Club. У обоих из них имеется масса достоинств, а также многочисленные постоянные клиенты. Сделать выбор

между этими двумя сервисами зачастую бывает достаточно сложно, поэтому необходимо выполнить их сравнение.

Оценив все плюсы и минусы обоих сервисов можно утверждать, что они взимают очень высокие для российского рынка комиссии, которые очень отягощают работу заведений и вынуждают их увеличивать цены на свою продукцию, что для малых заведений в итоге приводит либо к потери клиентов, либо к потери прибыли.

CRM (от англ. Customer Relationship Management) — это система управления взаимоотношениями с клиентами, которая представляет собой прикладное программное обеспечение предназначенное для организаций и нацеленное на автоматизацию стратегий взаимодействия с заказчиком в том числе для увеличения объемов продаж, повышения эффективности маркетинга и повышения качества обслуживания клиентов за счет накопления информации о них и истории взаимодействия, определения и улучшения бизнес-процессов с последующим анализом результатов.

На данный момент существует множество CRM предназначенных для автоматизации бизнеса в сфере общественного питания с возможностью настройки процесса доставки, вот некоторые из них:

- Poster;
- СБИС Presto;
- ЛТMeal;
- Супердоставка;
- Смартомато.

Как уже отмечалось выше, существует множество CRM, позволяющих автоматизировать процесс доставки еды, все они имеют свои недостатки и сильные стороны. Множество из них предоставляют пробные периоды использования или пробные версии с ограниченным функционалом, но версии с необходимым и достаточным функционалом для реальной организации бизнеса всегда являются платными. Поэтому в соответствии с результатами анализа бизнеса, который ведет предприятие и на основании финансовых расчетов следует выбирать оптимальный вариант в каждом отдельном случае в соответствии с необходимым для работы функционалом, ценой и качеством.

UML — это язык широкого профиля с открытым стандартом, который использует графические обозначения для создания абстрактных моделей (UML—

моделей) проектируемых систем, в том числе бизнес-процессов и организационных структур. При том, что он не является языком программирования, все же многие среды проектирования позволяют генерировать код на основе созданных при помощи него моделей [3].

UML представляет собой графическую нотацию предназначенную для моделирования и описания всех процессов, которые протекают в процессе разработки. Основу данного языка представляют диаграммы, различающиеся по типам и предназначенные для моделирования различных аспектов разработки [4].

Все диаграммы условно подразделяются на поведенческие и структурные. При этом поведенческие диаграммы отображают процессы, которые протекают в моделируемой среде. Структурные диаграммы призваны отображать элементы, из которых состоит система. Стоит заметить, что одни и те же типы диаграмм могут вполне использоваться для моделирования бизнес-процессов, а также для непосредственного проектирования архитектуры [3].

Диаграмма вариантов использования представляет собой отправную точку в процессе моделирования. Она предназначена для описания взаимодействий в проектируемой системе с любыми внешними и внутренними объектами (пользователями, системами и т.п.). Основными понятиями при работе с такой диаграммой являются Актёр «Actor» и Вариант использования «Use Case» [4].

Диаграмма последовательностей предназначена для расшифровывания последовательностей действий при процессе выполнения некоторого варианта использования. Иначе говоря, вариант использования дает ответ на вопрос «Что делает актер?», а последовательность дает ответ на вопрос «Как реагирует система в ходе выполнения этого варианта использования?» [5].

Sequence Diagram всегда берет свое начало с актера, который инициирует процесс. Сверху в диаграмме расположены элементы, классы, а также компоненты, задействованные в процессе работы. На самой диаграмме вариантов отмечаются линии жизни для каждого объекта и процесс взаимодействия, который в свою очередь обозначается стрелками. В терминах диаграмм последовательностей это взаимодействие называется «Сообщение» («Message») [5].

Диаграммы состояний, как правило, применяют с целью визуального

представления переходов своих состояний какого-либо элемента, например инстанцированного класса. При помощи этих диаграмм описывают, как состояния классов, так и поведение бизнес-объектов. Данные диаграммы выполняют вспомогательные функции и их применяют в дополнение к остальным диаграммам [4].

Элементы «Состояние» («State») представляют собой состояние рассматриваемого объекта, либо процесса в соответствующий момент времени. При этом название состояния представляет его описание [4].

В отличие диаграммы вариантов использования и диаграммы последовательностей, диаграмма классов имеет структурный характер. Она предназначена для изображения классов разрабатываемого программного обеспечения, а также взаимосвязей. Так же, как и указанные диаграммы ее возможно представлять в терминах конкретных классов, либо в терминах бизнес-объектов [6]. Данные диаграммы, как правило, заполняются одновременно с диаграммами последовательностей при процессе моделирования работы вариантов использования [4].

Класс должен состоять из двух частей — заголовков с именем класса, а также тела с описанием полей («Атрибуты» — в терминах UML) и методов («Операции» — в терминах UML) [7].

В результате проведенного анализа систем автоматизированной доставки еды из кафе и теоретических основ унифицированного языка UML были определены необходимые и достаточные для представления моделируемых подсистем UML-диаграммы.

Второй раздел «Разработка с применением UML-диаграмм системы автоматизированной доставки продукции кафе» посвящен определению требований к разрабатываемой системе, выбору программного инструмента для работы с UML, построению с его помощью необходимых и достаточных для представления моделируемых подсистем диаграмм, а также программной реализации подсистемы взаимодействия с пользователем.

Анализ требований заключается в первоначальном изучении идеи. Здесь важно определить, что клиент хочет получить, и как он представляет себе свой будущий продукт. Это делает возможным сформировать идею, а также оценить ее перспективы [1]. Затем исследуются конкуренты, аналогичные продукты представленные на рынке и целевая аудитория, определяются возможности и

проводится ряд исследований аналитического характера, которые позволяют понять, как превратить идею в действующий продукт [2].

Таким образом, при изучении идеи и её перспектив, а также аналитическом исследовании аналогов системы и целевой аудитории были выявлены следующие требования:

1. требования к функциональным характеристикам для подсистем взаимодействия: с пользователем, оператором, кассиром, кухней, курьером и администратором;
2. данные хранимые в базе данных;
3. требования к вводу/выводу данных с/на страницы web-сайта;
4. Требования к надежности;
5. Требования к составу и параметрам технических средств.

На основании проведенного анализа современных инструментов UML-моделирования, для решения поставленной задачи был выбран Visual Paradigm ввиду его достаточности и простого удобного интерфейса, а также доступности (30-дневная пробная версия), который представляет собой современный активно развивающийся продукт. Он в свою очередь обладает всеми необходимыми возможностями для анализа процессов, проектирования систем, кодогенерации в языки C++ и Java, а также в целом позволяет моделировать информационную систему бизнеса и процессов разработки. Здесь же стоит отметить такое преимущество Visual Paradigm перед остальными, как наличие большого количества справочной информации и обучающих курсов по работе с ним [8].

Поскольку диаграмма вариантов использования представляет собой отправную точку в процессе моделирования, то логично начать разработку комплекса UML-диаграмм именно с неё [6, 7]. Она предназначена для описания взаимодействий в проектируемой системе с любыми внешними и внутренними объектами (пользователями, системами и т.п.) [4].

В соответствии с требованиями задания были выделены все предполагаемые актеры и была построена диаграмма вариантов использования, которая затем была подвергнута декомпозиции в соответствии с выделенными подсистемами. При этом сами варианты использования агрегируются в блоках соответствующих подсистемам, которые представляют разрабатываемые модули системы, а актеры находятся вне этих подсистем.

Основываясь на выявленных требованиях и построенной диаграмме вариантов использования были описаны принципы работы системы и разработаны диаграммы последовательностей, состояний, а также классов для каждой из выделенных подсистем.

Выявленные при анализе требований данные хранимые в базе данных можно объединить в таблицы. Соблюдая требования всех первых трех нормальных форм, а также нормальной формы Бойса–Кодда эти таблицы были связаны и представлены в виде единой базы данных. При этом для всех полей сформированных таблиц выбраны типы данных в соответствии с хранимыми в них данными. Для строковых данных подобраны типы необходимой и достаточной длины.

В связи с возможностью тесного взаимодействия Visual Paradigm и языка программирования Java, для разработки программного кода был выбран именно этот язык в его последней 18-й версии. С другой же стороны для данного языка программирования существует мощнейший фреймворк Spring, который в свою очередь реализует такую концепцию в разработке приложений, как MVC. Здесь же стоит отметить, что в текущем проекте наиболее целесообразно использовать именно фреймворк, что наиболее подходит для разработки приложений средних масштабов в отличии от разработки на «чистом» языке, что используется для огромных проектов (недостаток — большая трудоёмкость и цена), или же применения CMS и CMF для малых проектов (недостаток — малая производительность и надёжность, а также ограничения разработчиков ввиду малой гибкости).

MVC (Model–View–Controller, т.е. «Модель–Вид–Контроллер») представляет собой схему разделения данных web–приложения и управляющей логики на отдельные компоненты: модель, представление и контроллер. Такой подход позволяет модифицировать и дополнять каждый из этих компонентов независимо, что дает превосходную гибкость. Модель является совокупностью данных, которые изменяются в ответ на команды контроллера. Представление отображает данные модели пользователю в ответ на изменение модели. А контроллер улавливает и интерпретирует воздействия пользователя, и оповещает модель о необходимости изменить свое состояние.

Также выполнив анализ современных СУБД была выбрана Microsoft SQL Server 2019 ввиду наличия бесплатной версии, что само по себе позволяет

опробовать её в условиях ведения бизнеса, и при необходимости перейти на платную версию в которой компенсировались выявленные в ходе использования недостающие характеристики. Для взаимодействия выбранной СУБД и фреймворка Spring целесообразно использовать Microsoft JDBC Driver разработанный специально для взаимодействия с Microsoft SQL Server, который позволяет использовать запросы к базе данных любой сложности.

Для дальнейшей программной реализации классов смоделированных на диаграмме классов удобно воспользоваться инструментом кодогенерации Visual Paradigm. Это позволит создать по указанному пути пустые шаблоны всех классов на выбранном языке программирования, которые остается лишь только разместить в соответствующих папках проекта и заполнить кодом алгоритмов.

Все классы моделей получают при помощи кодогенерации на данном этапе проектирования максимально полными, единственное, что необходимо для них сделать, это указать название пакета, в котором они содержатся, импортировать необходимые библиотеки, пометить класс аннотацией `@Entity` для указания что модель представляет собой сущность базы данных, а также задать поле со значением первичного ключа и стратегии его генерации при необходимости. Аналогичные дополнения инструкциями производятся и для всех остальных классов–моделей.

Для класса–контроллера необходимо также указать название пакета, к которому он принадлежит, необходимые для корректной работы импорты, созданные при кодогенерации поля следует пометить аннотацией `@Autowired` для автозаполнения инъекцией зависимости фреймворка Spring. Далее необходимо помечать каждый метод–обработчик аннотацией `@GetMapping()`, в скобках которой указывать строку с запросом, на который реагирует текущий метод, а также генерировать возможное исключение `SQLException` при работе с базой данных. Внутри метода в общем случае извлекаются передаваемые вместе с запросом параметры в объекте класса `Model` затем происходит обращение к базе данных за необходимой информацией методами драйвера JPA и/или драйвера JDBC (в случае более сложного запроса). Подобным образом заполняются остальные методы–обработчики, разница заключается лишь в строке SQL–запроса, который определяется контекстом исходного HTTP–запроса.

В завершение данного раздела можно отметить, что анализ требований позволил построить необходимые и достаточные для дальнейшей разработки UML–диаграммы, что в свою очередь сильно облегчило разработку базы данных и программную реализацию подсистемы взаимодействия с ней и пользователем.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной выпускной квалификационной работы была разработана система автоматизированной доставки продукции кафе на основе языка моделирования UML. При этом были исследованы особенности автоматизации в сфере общественного питания, произведен обзор и сравнение аналогичных систем, для которых в свою очередь были определены их достоинства и недостатки, сделаны выводы. Также был произведен обзор общих принципов построения диаграмм с помощью языка UML. Полученная при этом информация позволила решить поставленные задачи:

1. проведения анализа требований;
2. проектирования комплекса необходимых UML–моделей для дальнейшей разработки системы;
3. программной реализации алгоритмов подсистемы, которая отвечает за взаимодействие сервера с пользователем и базой данных, для оценки состоятельности спроектированного комплекса UML–моделей.

При проведении анализа были определены требования к функциональным характеристикам, данные которые должны храниться в базе данных, требования к вводу/выводу данных, а также к параметрам технических средств, что в дальнейшем позволило определить стек технологий, для программной реализации одной из подсистем.

При проектировании комплекса UML–моделей были синтезированы все необходимые и достаточные для последующей разработки системы диаграммы. А успешная программная реализация подсистемы взаимодействия с пользователем доказала состоятельность применения этих диаграмм.

Решение поставленных задач позволило добиться в полной мере главной цели работы, а именно, описать необходимую для ведения бизнеса систему автоматизированной доставки продукции кафе с применением на практике унифицированного языка моделирования UML. Что в свою очередь позволяет добиться упрощения взаимодействия исполнителя с заказчиком, а также внутри

главной команды исполнителей и/или сторонними командами исполнителей, в случае перераспределении объемов выполняемых работ. Это также позволяет облегчить дальнейший процесс написания исполняемого программного кода за счет возможности кодогенерации для выбранного языка программирования и обеспечить простоту релокации частей разрабатываемой системы на другие платформы. Релокация на другие платформы (например Android или iOS) в перспективе могла бы повысить конкурентоспособность данного бизнеса, так как сделала бы подсистемы взаимодействия более удобными в первую очередь для пользователей за счет покрытия большего количества разных устройств, и это неизбежно дало бы положительный толчок в развитии предприятия в целом.

Основные источники информации:

1. Организация доставки еды в ресторанах и кафе [Электронный ресурс]. — URL: <https://restoplace.cc/blog/organizaciya-dostavki-edu> (Дата обращения 01.05.2022). Загл. с экр. Яз. рус.
2. Автоматизация работы службы доставки [Электронный ресурс]. — URL: <https://vc.ru/services/71135-avtomatizaciya-raboty-sluzhby-dostavki-keys-pirogi-1> (Дата обращения 01.05.2022). Загл. с экр. Яз. рус.
3. Розенберг, Д. Применение объектного моделирования с использованием UML и анализ прецедентов : Пер. с англ. Слинкина А. / Д. Розенберг. — Москва: ДМК Пресс, 2012. — С. 160.
4. Буч, Г. Введение в UML от создателей языка. 2-е изд. : Пер. с англ. Мухин Н. / Г. Буч. — Москва: ДМК Пресс, 2015. — С. 496.
5. Фаулер, М. UML. Основы, 3-е изд. : Пер. с англ. Петухов А. / М. Фаулер. — Санкт-Петербург: Символ-Плюс, 2019. — С. 496.
6. Ларман, К. Применение UML и шаблонов проектирования, 2-ое изд. : Пер. с англ. / К. Ларман. — Москва: Вильямс, 2014. — С. 736.
7. Рэмбог, Д. UML 2.0. Объектно-ориентированное моделирование и разработка, 2-ое изд. : Пер. с англ. Питер Пресс / Д. Рэмбог. — Санкт-Петербург: Питер, 2017. — С. 544.
8. The №1 Development Tool [Электронный ресурс]. — URL: <https://www.visual-paradigm.com/> (Дата обращения 01.05.2022). Загл. с экр. Яз. рус.