

УДК 004.65

UDC 004.65

5.2.2. – Математические, статистические и инструментальные методы в экономике (физико-математические науки, экономические науки)

5.2.2 – Mathematical, statistical and instrumental methods in economics (physical and mathematical sciences, economic sciences)

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ОКАЗАНИЯ БЫТОВЫХ УСЛУГ

DEVELOPMENT OF AN INFORMATION SYSTEM FOR THE PROVISION OF HOUSEHOLD SERVICES

Рыбьянцева Мария Сергеевна
Кандидат экономических наук, доцент
РИНЦ SPIN-код: 7874-8981
email: Riban1@mail.ru
ФГБОУ «Кубанский государственный аграрный университет», 350044, Россия, г. Краснодар, ул. Калинина 13

Rybyantseva Maria Sergeevna
Candidate of Economic Sciences associate professor
RSCI SPIN-code: 7874-8981
email: Riban1@mail.ru
Kuban State Agricultural university, 350044, Russia, Krasnodar, Kalinina, 13

Ефимиади Леонид Константинович
Ассистент кафедры информационных систем
РИНЦ SPIN-код: 7180-6205
email: efimiadi01@mail.ru
ФГБОУ «Кубанский государственный аграрный университет», 350044, Россия, г. Краснодар, ул. Калинина 13

Efimiyadi Leonid Konstantinovich
Assistant of the Department of Information Systems
RSCI SPIN-code: 7180-6205
email: efimiadi01@mail.ru
Kuban State Agricultural university, 350044, Russia, Krasnodar, Kalinina, 13

Разработка веб-сервиса для оказания бытовых услуг позволит решить проблему нехватки времени на поиск надежного специалиста. Функциональность продукта даст возможность клиентам осуществлять поиск и выбор специалиста по категориям, а исполнителям – удобно размещать объявления и получать на них быстрый отклик. Веб-сервис также обеспечит гарантированную и фиксированную оплату услуги и возможность ведения рейтинга специалистов по оценкам реальных пользователей. В данной статье для выбранной предметной области выделены роли, Epic-задача, детализированы User Story для данного проекта для пользователя и клиента. Расписаны критерии приемки для регистрации, авторизации, отклика, принятия отклика, принятия условий, проверки и отзыва. Выделены Task-задачи для некоторых Story. Приведены примеры UseCase и шаблон для его описания для «Поиска исполнителя» и «Отклика на заявку клиента». Выделены 12 сущностей: «Роли», «Города», «Специальности», «Пользователи», «Операции оплаты», «Типы операций оплаты», «Заявки», «Рабочая область», «Отзывы», «Статусы заявок», «Статусы ставок», «Ставки». Описаны входные и выходные параметры. Приведен пример запроса и ответа

The development of a web service for providing household services will solve the problem of lack of time to find a reliable specialist. The functionality of the product will enable customers to search and select a specialist by category, and performers - to conveniently post ads and receive a quick response to them. The web service will also provide a guaranteed and fixed payment for the service and the ability to maintain a rating of specialists based on real users' assessments. In this article, roles and an Epic task are highlighted for the selected subject area, and User Stories for this project for the user and the client are detailed. Acceptance criteria for registration, authorization, response, response acceptance, acceptance of conditions, verification and review are described. Task tasks are highlighted for some Stories. Examples of UseCase and a template for its description are given for "Search for a performer" and "Response to a client's request". 12 entities are identified: "Roles", "Cities", "Specialties", "Users", "Payment transactions", "Payment transaction types", "Applications", "Workspace", "Reviews", "Application statuses", "Bid statuses", "Bids". Input and output parameters are described. An example of a request and response is given

Ключевые слова: БИЗНЕС-ТРЕБОВАНИЯ, USER CASE, USER STORY, СУЩНОСТИ, МОНОЛИТНАЯ АРХИТЕКТУРА

Keywords: BUSINESS REQUIREMENTS, USE CASE, USER STORY, ENTITIES, MONOLITHIC ARCHITECTURE

<http://dx.doi.org/10.21515/1990-4665-206-024>

<http://ej.kubagro.ru/2025/02/pdf/24.pdf>

Пользовательские истории (UserStory) – это инструмент, позволяющий трансформировать бизнес-требования в функциональные. UserStory описывает, что делает пользователь определенной роли для достижения результата, и что нужно сделать разработчику, чтобы воплотить эту задачу в жизнь. UseCase описывает поведение пользователя по шагам при взаимодействии с разрабатываемым продуктом.

Рассмотрим применение методов на примере веб-сервиса для оказания бытовых услуг. Разработка веб-сервиса для оказания бытовых услуг позволит решить проблему нехватки времени на поиск надежного высококвалифицированного специалиста. Функциональность продукта даст возможность клиентам осуществлять поиск и выбор специалиста по категориям, а исполнителям – удобно размещать объявления и получать на них быстрый отклик. Веб-сервис также обеспечит гарантированную и фиксированную оплату услуги и возможность ведения рейтинга специалистов по оценкам реальных пользователей.

В рамках данного проекта можно выделить 4 роли: зарегистрированный клиент – «Client», зарегистрированный специалист – «Specialist, незарегистрированный пользователь – «User» и администратор – «Admin». На данном этапе необходимо обозначить Epic-задачи для проекта. В нашем случае Epic будет один: «SHS-001 Разработка информационной системы для оказания бытовых услуг». Эпику (как и последующим типам задач) присвоен код SHS (Service for Household Services) для определения его предметной области.

Выделим Story для данного проекта:

1. «SHS-002 Регистрация пользователя».
2. «SHS-003 Авторизация пользователя».
3. «SHS-004 Поиск специалиста».
4. «SHS-005 Создание заявки».

5. «SHS-006 Назначение специалиста».
6. «SHS-007 Оплата услуги».
7. «SHS-008 Отзыв и оценка услуги».
8. «SHS-009 Подача резюме».
9. «SHS-010 Отзыв и оценка клиента».
10. «SHS-011 Отклик на заявку».

Опишем выделенные Story по технике UserStory. У незарегистрированного пользователя ограничена работа с функциональностью программы, поэтому для него может быть описана только одна User Story (таблица 1).

Таблица 1 – User Story пользователя

Id	As a <type of user>	I can <functionality>	so that <rationale>
Ключевая функция 1: Действия с учетной записью			
User	Пользователь	Зарегистрироваться в системе	Для дальнейшей работы с сайтом

В таблице 2 приведен перечень User Stories для клиента.

Таблица 2 – User Stories для клиента

Id	As a <type of user>	I can <functionality>	so that <rationale>
Ключевая функция 1: Действия с учетной записью			
Client	Зарегистрированный Клиент	Авторизоваться в системе	Для получения доступа к сайту
Ключевая функция 2: Получение информации о специалистах			
Client	Зарегистрированный Клиент	Осуществить поиск специалиста определенной категории	Для просмотра имеющихся специалистов
Ключевая функция 3: Запрос услуги			
Client	Зарегистрированный Клиент	Подать заявку о необходимой услуге	Для привлечения специалистов
Client	Зарегистрированный Клиент	Назначить специалиста	Для выполнения заказа
Ключевая функция 4: Оплата услуги			
Client	Зарегистрированный Клиент	Оплатить услугу	Для завершения и принятия работы специалиста
Ключевая функция 5: Оценка услуги			
Client	Зарегистрированный Клиент	Выставить оценку по 5-звездочной шкале	Для изменения рейтинга специалиста на сайте
Client	Зарегистрированный Клиент	Написать отзыв о работе мастера	Для отражения обратной связи о выполненной работе

В качестве примера опишем критерии приемки с использованием синтаксиса GHERKIN (таблица 3).

Таблица 3 – Примеры критериев приемки

Позиция	Критерии приемки
Регистрация	Scenario: Registration Given the user is on the login page When the user entered "Surname" And the user entered "Name" And the user entered "City" And the user entered "Role Client" And the user entered " Phone number " And the user entered "Password" Then the successful registration message should display
Авторизация	Scenario: Authorization Given the user is on the login page When the user entered " Phone number " And the user entered "Password" Then the user goes to user page
Отклик	Scenario: Response Given the user is on the order page When the user clicked on button "Respond" And the user entered "Cost" And the user entered text on the text field Then the response appears on the order page
Принятие отклика	Scenario: Approval of conditions Given the user is on the order page When the user clicked on button "Get bet" Then the revoke will be getting
Принятие условий	Scenario: Acceptance of conditions Given the user is on the work page When the user clicked on button "Accept the Terms" Then the conditions will be accepting
Проверка	Scenario: Check Given the user is on the order page When the user clicked on button "Order completed" Then the order status changes to complete And Message with a proposal to revoke should display
Отзыв	Scenario: Revoke Given the user is on the specialist page When the user clicked on button "Revoke" And the user entered text in entry field And the user clicked on button "Post" Then the message send revoke should display And the revoke appears on the specialist page

Task-задачи отражают конкретные работы команды в рамках одной Story. Выделим Task для некоторых Story:

Story 3 «SHS-004 Поиск специалиста»:

- Task 1: «SHS-020 Разработка интерфейса поиска с фильтрами»;
- Task 2: «SHS-021 Настройка серверного API для обработки запросов на поиск»;
- Task 3: «SHS-022 Настройка отображения результатов поиска»;
- Task 4: «SHS-023 Реализация сортировки результатов поиска».

Story 4 «SHS-005 Создание заявки»:

- Task 1: «SHS-024 Разработка формы для создания заявки»;
- Task 2: «SHS-025 Реализовать валидацию данных в клиентской части»;
- Task 3: «SHS-026 Настройка серверного API для обработки запросов на создание заявки»;
- Task 4: «SHS-027 Реализация сохранения заявки в базе данных».

Story 5 «SHS-006 Назначение специалиста»:

- Task 1: «SHS-028 Разработка интерфейса для выбора специалиста»;
- Task 2: «SHS-029 Настройка серверного API для обработки запросов на назначение специалиста»;
- Task 3: «SHS-030 Обновление статуса заявки в базе данных»;
- Task 4: «SHS-031 Реализация функционала отражения информации о назначенном специалисте в заявке»

Story 6 «SHS-007 Оплата услуги»:

- Task 1: «SHS-032 Разработка интерфейса для выбора метода оплаты»;
- Task 2: «SHS-033 Настройка серверного API для обработки платежей»;
- Task 3: «SHS-034 Настройка валидации данных платежа»;
- Task 4: «SHS-035 Разработка страницы с деталями транзакции».

Story 8 «SHS-009 Подача резюме»:

- Task 1: «SHS-040 Разработка формы подачи резюме»;
- Task 2: «SHS-041 Настройка серверного API для обработки резюме»;
- Task 3: «SHS-042 Реализация возможности загрузки резюме в формате PDF»;
- Task 4: «SHS-043 Реализовать уведомление об изменении статуса рассмотрения резюме».

Story 9 «SHS-010 Отзыв и оценка клиента»:

- Task 1: «SHS-044 Разработка интерфейса для оставления отзыва и оценки о клиенте»;
- Task 2: «SHS-045 Настройка серверного API для обработки отзывов и оценок специалистов»;
- Task 3: «SHS-046 Настройка модерации отзывов»;
- Task 4: «SHS-047 Разработка страницы с отображением отзывов и оценок на странице клиента».

Story 10 «SHS-011 Отклик на заявку»:

- Task 1: «SHS-048 Разработка интерфейса для отклика на заявку»;
- Task 2: «SHS-049 Настройка серверного API для обработки откликов специалистов»;
- Task 3: «SHS-050 Обновление статуса заявки в базе данных»;
- Task 4: «SHS-051 Разработка функционала просмотра откликов специалистов».

Варианты использования можно выделить на основании описанных ранее Story (1 Story = 1 вариант использования).

Приведем некоторые примеры UseCase и шаблон для его описания:

- «Поиск исполнителя» (таблица 4);

Альтернативный поток 1 имеет особенности:

- 1) начинается после шага 3 основного потока;
- 2) система выдает сообщение, что таких специалистов на портале нет.

Таблица 4 – UseCase «Поиск исполнителя»

Код и Наименование ВИ	Client2. Поиск исполнителя
Краткое описание	Данный вариант использования позволяет клиенту осуществлять поиск специалиста по оказанию услуг на сайте. Для этого ему необходимо ввести в поле поиска название интересующей услуги и нажать на кнопку «Поиск».
Актеры Главный (инициирует ВИ) Второстепенный	Клиент
Ссылки на другие ВИ	-
Предусловия	Необходимо найти список подходящих мастеров
Основной поток	1) Клиент отправляется в меню «Специалисты» 2) Клиент выставляет фильтры 3) Клиент нажимает кнопку поиска 4) Система выдает пользователю подходящих специалистов
Постусловия	Клиент видит перед собой список из специалистов с учетом выставленных фильтров
Альтернативные потоки	Нет подходящего мастера
Альтернативный поток 1	Система после поиска не вывела ни одного специалиста

– «Отклик на заявку клиента» (таблица 5).

Таблица 5 – UseCase «Отклик на заявку клиента»

Код и Наименование ВИ	Specialist3. Отклик на заявку клиента
Краткое описание	Данный вариант использования помогает специалисту получить работу. Мастер откликается на услугу при помощи кнопки «Отклик», что дает ему возможность стать исполнителем заказа.
Актеры Главный (инициирует ВИ) Второстепенный	Специалист
Ссылки на другие ВИ	-
Предусловия	Специалист желает выполнить заказ
Основной поток	1) Специалист в поиске заказов по профилю выбирает конкретный заказ 2) Специалист откликается на заказ при помощи кнопки
Постусловия	Отклик будет отображен у клиента
Альтернативные потоки	1) Нет ни одного подходящего заказа по конкретному профилю
Альтернативный поток 1	Нет ни одного подходящего заказа по конкретному профилю. Начинается после 1 шага основного потока. Система выдает сообщение о том, что заказов в данный момент нет

Выбор архитектуры зависит от множества факторов: требования к масштабируемости, гибкости, устойчивости, технологической независимости, простоты реализации и т.д. Необходимо взвесить эти факторы перед принятием решения о выборе вида архитектуры и обосновать, какие факторы при этом для нас являются наиболее важными.

Для веб-сервиса по оказанию бытовых услуг на начальных этапах (в качестве начального решения) целесообразно использовать монолитную архитектуру, которая представляет собой единое приложение, где все компоненты связаны и разрабатываются вместе. Основными преимуществами такой архитектуры являются: простота разработки и развертывания, упрощенное тестирование и управление.

Несмотря на ограничения масштабируемости и гибкости, монолитная архитектура является хорошим выбором для старта, поскольку может покрыть все поставленные первичные требования и обозначенные объемы работ.

По мере развития проекта, в случае его успеха на рынке, может потребоваться переход на более сложные и прогрессивные виды архитектур (например, на микросервисную).

Необходимо описать сущности (для монолитной архитектуры) или микросервисы (для микросервисной архитектуры), которые будут реализованы в системе.

Для монолитной архитектуры представить сущности можно в виде таблиц базы данных.

Для микросервисной архитектуры необходимо указать стрелками виды интеграционных взаимодействий (REST, Kafka и тд).

Веб-сервис по оказанию бытовых услуг состоит из 12 сущностей: «Роли», «Города», «Специальности», «Пользователи», «Операции оплаты», «Типы операций оплаты», «Заявки», «Рабочая область», «Отзывы», «Статусы заявок», «Статусы ставок», «Ставки».

На диаграмме показано, как все сущности связаны друг с другом, и обозначены первичные и вторичные ключи (рисунок 1).

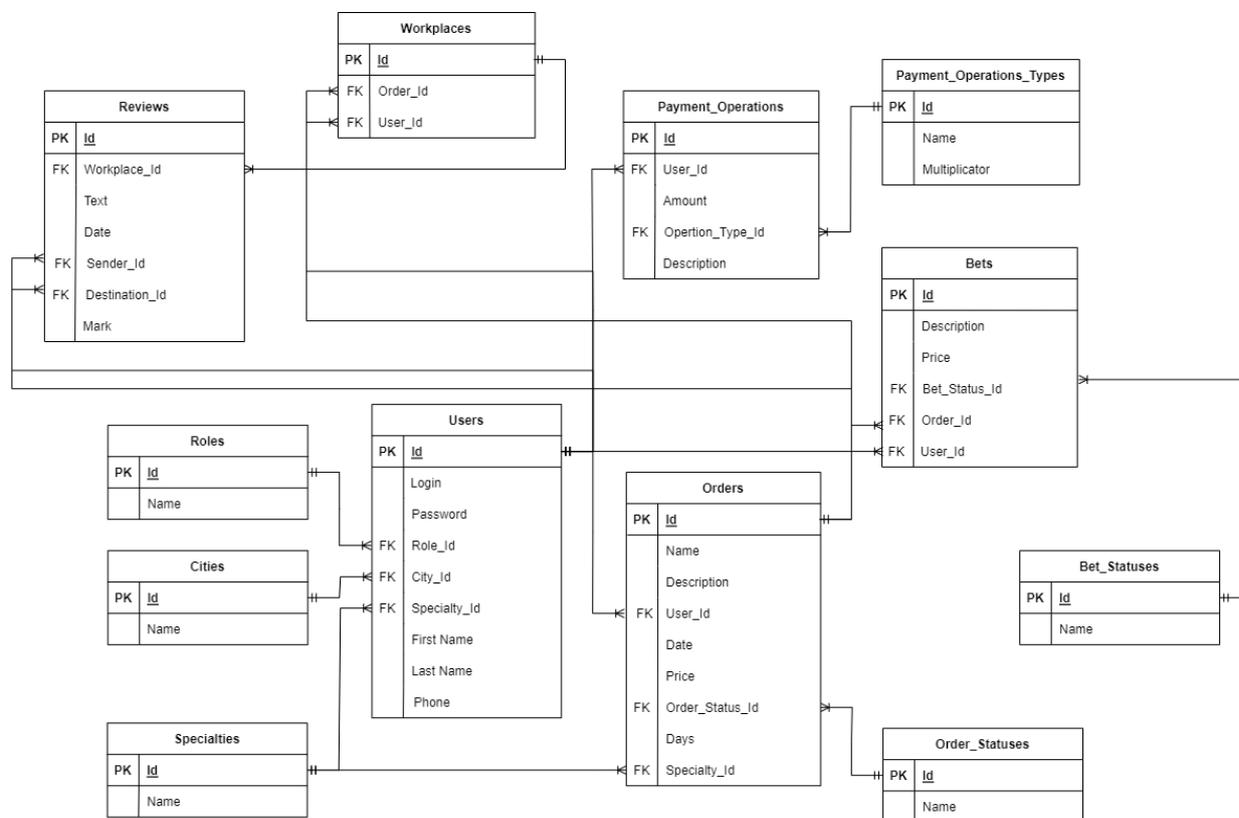


Рисунок 1 – Сущности веб-сервиса

В качестве интеграций может быть выбрано любое взаимодействие по REST, SOAP, Kafka, RabbitMQ и тд.

Например, может быть описано взаимодействие между разрабатываемой системой и смежным продуктом (для монолитной архитектуры) и между микросервисами (в рамках микросервисной архитектуры).

В качестве примера интеграции рассмотрим взаимодействие платформы для оказания бытовых услуг с платежным сервисом для проведения оплаты.

Описание входных параметров приведено в таблице 6.

Таблица 6 – Описание входных параметров

Номер	Параметр	Тип данных	Обязательность	Описание	Пример
1	orderId	string	Да	Уникальный идентификатор заказа	45453534
2	amount	float64	Да	Сумма платежа	1500.00
3	currency	string	Да	Валюта платежа	RUB
4	paymentMethod	string	Да	Метод оплаты	credit_card
5	cardDetails	obj	Нет	Реквизиты карты	
6	cardDetails. cardNumber	string	Нет	Номер карты	47640912 97808344
7	cardDetails. expiryDay	string	Нет	Срок действия карты	07/2029
8	cardDetails.cvc	string	Нет	CVC-код карты	426
9	description	string	Нет	Описание	Оплата за уборку

Описание выходных параметров приведено в таблице 7.

Таблица 7 – Описание выходных параметров

Номер	Параметр	Тип данных	Обязательность	Описание	Пример
1	transactionId	string	Да	Уникальный идентификатор транзакции	abc78490
2	status	string	Да	Статус платежа	success
3	message	string	Да	Сообщение о результате транзакции	Платеж успешен
4	amount	float64	Да	Сумма списания	1500.00
5	currency	string	Да	Валюта платежа	RUB
6	createdDate	string	Да	Дата и время транзакции	2024-10-01T10:00:00Z

Рассмотрим пример запроса:

POST /api/payments/create

Content-Type: application/json

{

"orderId": "45453534",

```
"amount": 1500.00,  
"currency": "RUB",  
"paymentMethod": "credit_card",  
"cardDetails":  
{  
  "cardNumber": "4764091297808344",  
  "expiryDate": "07/2029",  
  "cvc": "426"  
},  
"description": "Оплата за уборку"  
}
```

Рассмотрим пример ответа:

```
{  
  "transactionId": "abc78490",  
  "status": "success",  
  "message": "Платеж успешен",  
  "amount": 1500.00,  
  "currency": "RUB",  
  "createdDate": "2024-10-01T10:00:00Z"  
}
```

Таким образом, рассмотрено применение методов Use Case и User Story для веб-сервиса для оказания бытовых услуг. Выделены 12 сущностей для выбранной предметной области, описаны входные и выходные параметры.

Список использованной литературы:

1. Разработка требований к программному обеспечению / К. Вигерс, Д. Битти. – СПб : БХВ, 2024. – 736 с.

References:

1. Razrabotka trebovanij k programmnomu obespecheniju / K. Vigers, D. Bitti. – SPb : BHV, 2024. – 736 s.