1. Документация Введение [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B1%D0%B8%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5>
2. Документация Введение [Электронный ресурс]. – Режим доступа: <https://artismedia.by/blog/plyusy-i-minusy-mobilnyh-sajtov-i-prilozhenij/>
3. https://www.sites.google.com/site/anisimovkhv/learning/pris/lecture/tema15/tema15\_2
4. Леоненков, А.В. Объектно-ориентированный анализ и проектирование с использованием UML / А.В. Леоненков. – [www.intuit.ru](http://www.intuit.ru/).

<http://www.telenir.net/uchebniki/samouchitel_uml/p5.php>

2.3 Диаграмма классов

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы.

5 https://www.nazametku.com/dlia-raboty/idef0-metodologiya-notaciya-principy-model/#:~:text=%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0

4 Тестирование программного продукта

4.1 Виды тестирования

Тестирование программного обеспечения (ПО) - это процесс исследования ПО с целью выявления ошибок и определения соответствия между реальным и ожидаемым поведением ПО, осуществляемый на основе набора тестов, выбранных определённым образом. [8]

**С целью тестирования является:**

1. Повысить вероятность того, что приложение, предназначенное для тестирования, будет работать правильно при любых обстоятельствах.
2. Повысить вероятность того, что приложение, предназначенное для тестирования, будет соответствовать всем описанным требованиям.
3. Предоставление актуальной информации о состоянии продукта на данный момент. [9]

**Уровни тестирования:**

Модульное тестирование- это процесс исследования ПО, при котором тестируется минимально возможный компонент, например, отдельный класс или функция. Часто модульное тестирование осуществляется разработчиками программного обеспечения.

Интеграционное тестирование- это процесс исследования ПО, при котором тестируется интерфейсы между компонентами или подсистемами.

Системное тестирование- это процесс исследования ПО, при котором тестируется интегрированная система на её соответствие требованиям заказчика. Альфа и Бета тестирование относятся к подкатегориям системного тестирования.

Альфа-тестирование- это процесс имитации реальной работы разработчиков с программным продуктом, или реальная работа потенциальных пользователей с системой.

Бета-тестирование- это распространение версий с ограничениями для некоторой группы лиц, с целью проверки содержания допустимо минимального количества ошибок в программном продукте.

Функциональное тестирование- тестирование ПО, направленное на проверку реализуемости функциональных требований. При функциональном тестировании проверяется способность ПО правильно решать задачи, необходимые пользователям.

Тестирование чёрного ящика (black box) - тестирование ПО, при котором тестировщик имеет доступ к ПО только через интерфейсы заказчика, либо через внешние интерфейсы, позволяющие другому компьютеру или процессу подключиться к системе для тестирования.

Тестирование белого ящика (white box)- тестирование ПО, при котором тестировщик имеет доступ к исходному коду программы и может писать код, связанный с библиотеками тестируемого ПО. Основной трудностью является сложность отслеживания вычислений времени выполнения.

Регрессионное тестирование (regression testing)- тестирование ПО, при котором проводится проверка ранее найденных ошибок, а также проверка основной функциональности. Проводится, как правило, на каждой новой версии программного продукта. Регрессивное тестирование является наиболее важной фазой тестирования непосредственно перед окончанием работ над продуктом, так как непосредственно перед релизом продукта крайне необходимо проверить не только основную функциональность, но и то, что ни одна из ранее найденных ошибок не повторяется в финальной версии.

Автоматизированное тестирование (automated testing) - тестирование при котором используются программные средства для выполнения тестов и проверки результатов выполнения. Автоматизированное тестирование, несомненно, приносит пользу и экономит время и ресурсы компании. [9]

4.2 Test-Case

Test Case — это профессиональная документация тестировщика, последовательность действий направленная на проверку какого-либо функционала, описывающая как придти к фактическому результату.

Каждый тест кейс должен иметь 3 части:  
PreConditions — это список действий, которые приводят систему к состоянию пригодному для проведения основной проверки. Либо список условий, выполнение которых говорит о том, что система находится в пригодном для проведения основного теста состояния.  
Test Case Description — это список действий, переводящих систему из одного состояния в другое, для получения результата, на основании которого можно сделать вывод о удовлетворении реализации, поставленным требованиям.

PostConditions — это список действий, переводящих систему в первоначальное состояние.

Test Case Запись на прокат (см. Таблица 4.1)

Таблица 4.1

|  |  |
| --- | --- |
| Уникальный идентификатор и краткое имя | А1 Запись на прокат. |
| Описание задачи | Клиент записывается на прокат. |
| Предварительные условия | PRE-1. В данной момент есть свободная лошадь. |
| Выходные условия | POST-1. Лошадь забронируется. |
| Список действий | * Клиент выбирает тренера; * Система заносит информацию (дата, время, количество лошадей) о записи на прокат в базу данных. |
| Исключения | Ex-1. Нет свободных лошадей.   * Клиенту выдается сообщение «В данный момент нет свободных лошадей». |

Test Case Запись на прокат (см. Таблица 4.1)

Test Case Запись на тренировку (см. Таблица 4.2)

Таблица 4.2

|  |  |
| --- | --- |
| Уникальный идентификатор и краткое имя | А2 Запись на тренировку. |
| Описание задачи | Клиент записывается на тренировку . |
| Предварительные условия | PRE-1. В приложении есть все курсы к которому вы хотели записаться. |
| Выходные условия | POST-1. Запись заносится в базу. |
| Список действий | * Клиент выбирает курс; * Клиент выбирает дату и время; * Клиент выбирает тренера; * Клиент выбирает лошадь; * Нажимает на кнопку «Записаться»; |
| Исключения | Ex-1. Если запись на этот день есть.   * Клиенту выдается сообщение «Вы уже записались на этот день». |

Test Case Запись на тренировку (см. Таблица 4.2)

4.3 Модульное тестирование