**Аттестация на уровень 3 квалификации младший специалист по качеству ПО**

**Теоретическая часть: продолжительность 2 часа.** С 11:00 до 13:00

**Критерии успешности прохождения теоретической части:**

Необходимо по теоретической части набрать 20 баллов.

Задание №1 (выполняется письменно)

Необходимо провести функциональное тестирование web-приложения, представляющего из себя online-магазин, на котором зарегистрированные пользователи осуществляют покупки. Основные компоненты приложения — регистрация, логин, каталог товаров, поиск товаров, корзина товаров, оплата покупки. Разработка приложения будет проводится в несколько итераций, на каждой из которых в каждый из компонентов приложения будут добавляться новый функционал. По окончании разработки предполагается поддержка приложения компанией разработчиком.

Письменно, основываясь на вышеизложенной информации о приложении, выполнить следующие задания:

1. Написать план тестирования данного приложения. План тестирования должен включать:
   1. Типы тестирования, которые будут произведены.
   2. Требования к тестовому окружению.
   3. Список дополнительных инструментов, которые будут необходимы при тестировании.
2. Подходы (что будет тестироваться), которые необходимо использовать для тестирования каждого из компонентов и инструменты.
3. Что должно быть сделано в системе управления проектом, до того как вы приступите к тестированию.

Критерии оценки: **Максимум 25 баллов за задание. Количество набранных вами баллов будет зависеть от того, насколько полно вы раскроете процесс тестирования описанного в задании веб-приложения.**

Таблицу необходимо убрать из раздаточного материала

|  |  |  |
| --- | --- | --- |
| Вопросы: | Инструменты и подходы. | Баллы |
| 1. Типы тестирования | Функциональное, нагрузочное, безопасность, удобство использования.  Тестировавание сопровождаемости и переносимости.  Виды тестирования, соответствующие метрикам качества. | 1 балл  1 балл  1 балл |
| 1. Требования к тестовому окружению. | Расписанная архитектура тестового стенда. | 1 балл |
| 1. Дополнительные инструменты. | Генератор нагрузки, генератор данных, система автоматизированного тестирования (Selenium), инструмент доступа к back-end, к консоли БД, плагины к браузерам для мониторинга http запросов. Sandbox платежной системы. | 1 балл |
| 1. Указано, что тестируем login. | Ручное тестирование | 1 балл |
| 1. Указано, что тестируем login. | Использование инструментов автоматизированного тестирования с обоснованием (Selenium). | 1 балл |
| 1. Указано, что тестируем login. | Указано, что прежде чем начать тестирование, данные должны быть сгенерированы. | 1 балл |
| 1. Указано, что тестируем регистрацию пользователей. | Ручное тестирование | 1 балл |
| 1. Указано, что тестируем регистрацию пользователей. | Использование инструментов автоматизированного тестирования с обоснованием (Selenium). | 1 балл |
| 1. Указаны подходы тестирования каталога товаров. | Ручное тестирование. | 1 балл. |
| 1. Указаны подходы тестирования каталога товаров. | Указано, что прежде чем начать тестирование, данные должны быть сгенерированы. | 1 балл. |
| 1. Указаны подходы тестирования каталога товаров. | Использование инструментов автоматизированного тестирования с обоснованием (Selenium). | 1 балл |
| 1. Указано, что тестируем оплату. | Ручное тестирование. | 1 балл |
| 1. Указано, что тестируем оплату. | Sandbox используемой платежной системы. | 1 балл |
| 1. Указаны подходы тестирования поиска в каталоге товаров. | Ручное тестирование | 1 балл |
| 1. Указаны подходы тестирования каталога товаров. | Использование инструментов автоматизированного тестирования с обоснованием (Selenium). | 1 балл |
| 1. Указаны подходы тестирования каталога товаров. | Указано, что прежде чем начать тестирование, данные должны быть сгенерированы. | 1 балл |
| 1. Указано тестирование верстки |  | 1 балл |
| 1. Нагрузочное тестирование | Указано тестирование и используемые инструменты. | 1 балл |
| 1. Нагрузочное тестирование | Указаны виды нагрузочного тестирования. | 1 балл |
| 1. Тестирование usability. | Указаны критерии usability, которые приемлемы для данного проекта. | 1 балл |
| 1. Тестирование безопасности. | Указаны критерии и что будет протестировано. | 1 балл |
| 1. Исследование возникающих ошибок в back-end | Доступ к MySQL сервер (консоль или клиентское приложение), доступ к логам. | 1 балл |
| 1. Тестирование дополнительных элементов, которые могут присутствовать в приложении, к примеру, админка, которые явно не указаны в задании, но обязательно присутствуют. | Ручное или автоматизированное тестирование. | 1 балл. |

**Максимум 25 баллов за задание.**

**Задание № 2**

Сформировать MySQL запрос для TCC базы данных, показывающий процентное соотношение различных API запросов за сутки. Результат сохранить в csv файле в каталоге пользователя, который выполнил вход в систему.

**Выполненное задание оценивается в 5 баллов.**

Задание № 3

Определить, какие методы HTTP поддерживает сервер [www.myconnectedsite.com](http://www.myconnectedsite.com) . Использовать любой возможный инструмент. Описать решение задания. Screenshot с ответом прикрепить к redmine.

**Выполненное задание оценивается в 5 баллов.**

**Практическая часть аттестации:**

Продолжительность — 4 часа. С 14:00 до 18:00.

Задание выложить в редмайн, работа будет вестись в Джире.

Практическое задание на рабочем месте.

Необходимо составить тест-дизайн для следующего функционала проекта 3DW <https://issues.apdbox.com/browse/TTD-125>. Тест-дизайн должен включать в себя список тест-кейсов и дизайн каждого из тест-кейсов.

Задание будет оцениваться по:

* полноте тестового покрытия,
* по использованию техник тестирования по методу «черного ящика» .
* Соответствие принятому формату оформления тест-кейсов / тест-сьютов.
* английский язык (в описании компонента) согласно правилам, установленным в соответствующих минимумах на 2014г.

Условия выполнения: все данные тест-дизайна должны быть записаны в jira к установленному времени. Изменения после этого времени не принимаются. После выполнения задача должна быть назначена на К.Фирсанова для ревью.

Контрольный список тест-кейсов

1. Preconditions: creation of the required number of entities, collections and users.
2. Subscribe to collection.
3. Subscribe to entity.
4. Update entity and get notification.
5. Update collection and get notification.
6. Delete entity and get notification.
7. Delete collection and get notification.
8. Undelete entity and get notification.
9. Undelete collection and get notification.
10. Get subscriptions for current user.
11. Unsubscribe to entity.
12. Unsubscribe to collection.
13. Delete notification from entity and check, that it has been deleted.
14. Undelete notification from entity and check, that it has been deleted.