**Перший рівень** — відпрацюй навички на базовому рівні.

1. Склади порівняльну таблицю трьох видів тестової документації:

* Назва тестової документації:

1.Чек Ліст

* Основні характеристики:

-Пункти чек-листа містять лише назву перевірки, яку потрібно виконати, але без детального опису, як саме це зробити.

-Зазвичай чек ліст є не таким детальним як тест кейс, який в свою чергу більш детально описує якийсь певний тестовий випадок.

-Чек ліст потрібен для того щоб зберігати звітність та результати тестування.

-Чек ліст має бути чітким і структурованим, наприклад: змінити на телефоні яскравість за допомогою шторки і змінити яскравість з допомогою функції «автояскравість» - це різні речі

-Хороший чек ліст - це документ який описує що має бути протестовано.

**Переваги:**

-Використання чек лістів сприяє структуруванню інформації у співробітника

-Значна економія часу, при умові що чек лісти складені правильно

-Можливість використання чек ліста повторно з огляду на актуальність інформації яка міститься у ньому.

-При правильному записі необхідних дій у співробітника з'являється однозначне розуміння завдань. Це сприяє підвищенню швидкості навчання нових співробітників.

-Полегшує контроль виконання завдань, мається на увазі, що можна легко перевірити виконану роботу.

**Недоліки:**

-Якщо чек ліст складено не правильно, це може призвести до непорозуміння в команді і як наслідок втраті часу для виправлення дії яку ми виконували в тому чи іншому чек лісті.

-Чек-листи це форма прокрастинації. Можна довгий час складати список справ, але так і не розпочати його виконання.

* Назва тестової документації:

2. Тест - кейс

* Основні характеристики:

-Це детальний опис дій для перевірки якоїсь певної частини функціоналу, з чітким критерієм успішного проходження

-Тест кейс складається з 3-ох основних частин:

PreConditions (Передумови) – або список кроків, які приводять систему, що перевіряється, у стан, придатний для тестування, або список перевірок умов того, що система вже знаходитися в необхідному стані.

Test Case Description (Опис тестового випадку) – список дій, за допомогою яких здійснюється основна перевірка функціоналу (після якої і порівнюється фактичний результат із очікуваним).

PostConditions (Післяумови) – список дій, які повертають систему в початковий стан.

**Переваги:**

-Тест-Кейси можна довірити новачкові в команді, тому що вони не складні для сприйняття навіть для нової людини на проекті, при умові якщо частина функціоналу яку треба перевірити чітко окреслена

**Недоліки:**

-Тест кейси можуть бути дуже схожі один на одного, а перші кроки у перевірці одинаковими, внаслідок цього може виникнути копіпаст.

-Тест кейси швидко стають неактуальними, потрібно підтримувати їхню актуальність, наприклад: змінився лінк на сайт, нам потрібно зайти на тест кейс і обновити його, а таких тест кейсів може бути багато і тут спрацює людських фактор, десь можна пропустити змінити цей самий лінк.

саме тому тест кейси потрібно писати чітко і без повторень, щоб в подальшому це не призвело до копіпасту.

* Назва тестової документації:

3. Користувацький сценарій

* Основні характеристики:

Користувацький сценарій (User Scenario) — це схема, яка дозволяє визначити, чому покупці опиняються на сайті і як реалізують свої плани за допомогою вашого продукту.

Найчастіше такі сценарії створюються у форматі коротких розповідей про певного еталонного користувача, мета якого — задовольнити свою потребу за допомогою вашого сайту чи додатку.

Сценарії пишуться до етапу техробіт зі створення сайту/програми, вони сфокусовані на діях та потребах користувачів. Технічним аспектам (стекам, інструментам, інтерфейсу) в цьому процесі немає місця.

**Переваги:**

-Дають можливість краще зрозуміти потреби користувачів

-Дозволяє аналізувати досвід користувачів безпосередньо під час запитів

-Дозволяє спроєктувати інтерфейс так, щоб він відповідав традиційним для клієнтів патернам і дозволяв досягти цілі, витративши мінімум сил та часу.

**Недоліки:**

-тестування тицяє вас носом у ваші ж помилки (це дуже неприємно);

-дуже вірогідно, що доведеться кавалок роботи переробити, а робити зовсім не хочеться;

-ви тестували не на персоні (представнику ЦА), тому є нюанси.

2. Склади чек-ліст для перевірки головної сторінки свого улюбленого інтернет-магазину (rozetka.ua, hotline.ua, silpo.ua тощо). Темплейт для чек-ліста можна взяти в кроці три теорії.

[Чек Ліст до Д/з 4](https://docs.google.com/spreadsheets/d/1woQnfKkYyO9AsiVBbmUzv8FM2d6zxHiJVuywiQ5GrNc/edit#gid=0)

**Другий рівень — детальніше заглибся в практику.**

1. Виконай завдання попереднього рівня.

2. Склади 5 позитивних і 5 негативних тест-кейсів для перевірки основної функціональності твого улюбленого інтернет-магазину (rozetka.ua, hotline.ua, silpo.ua тощо)

[Тест Кейси ДЗ 4](https://docs.google.com/spreadsheets/d/1d8p1p3-iwpK8jHHfrQNjl9rhjilovb7huIwNLsbN64s/edit#gid=0)

Прикріпив баг репорт до 6 тест кейсу

не знаю чи правильно написаний)

Прикол в тому що в мене дійсно не працює голосовий пошук на розетці)))

BUG REPORT

| Заголовок (Summary) | На сайті Rozetka не працює голосовий пошук, при безпосередньому натиску на поле «мікрофону» програма ніяк не реагує |
| --- | --- |
| Проєкт (Project) | Сайт інтернет магазину Rozetka |
| Компонент програми (Component) | Меню пошуку на головній сторінці, безпосередньо голосовий пошук |
| Номер версії (Version) | Rozetka version (16614.3.7.1.7, 16614) |

| Критичність (Severity) | Критичний (Critical) |
| --- | --- |
| Пріоритет (Priority) | Високий (High) |
| Статус (Status) | Відкритий (Open) |
| Автор (Author) | Denys Kizlyk |

| Призначений для (Assigned To) | Daria Samilyk |
| --- | --- |
| Опис (Description) | 1. операційна система; Mac OS Big Sur 11.7.2 (20G1020), Browser; Safariversion 16.2 (16614.3.7.1.7, 16614)  2. Відкрити сайт <https://rozetka.com.ua/ua/notebooks/c80004/preset=game/>  У Головному меню перейти до шапки пошукового меню, ліворуч від зеленої кнопки «Знайти» натискаємо на ярлик у формі мікрофона  3. Нічого не відбувається, програма ніяк не реагує на дії  4. Запуск віконця із дозволом використовувати вбудований мікрофон пристрою(ноутбуку) |
| Прикріплений файл (Attachment) | Прикріплюю скрін для відтворення проблеми |

**Третій рівень — різнобічно опануй тематику уроку.**

1. Виконай завдання двох попередніх рівнів.

2. Ти – засновник/ця стартапу, який планує випустити на ринок мобільний застосунок для обміну світлинами котиків для iOS та Android пристроїв. Стартап має команду з 3 розробників (back-end, iOS та Android). Вимоги та сценарії використання надаєш ти як головний стейкхолдер.

Завдання: Створи тест-план, який дозволить випустити продукт належної якості.

Document Revision History

| Date | Version | Description | Author | Reviewer | Approver |
| --- | --- | --- | --- | --- | --- |
| 27.01 | 0.1 | Test plan was created | D. Kizlyk |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**INTRODUCTION**

Customer wants a perfect application, which passed the full cycle of manual testing. Given the specificity of the application it is very important to have the same quality and the programm.

The Test Plan has been created to facilitate communication within the team members. This document describe approaches and methodologies that will apply to the unit, integration and system testing of the ” My Mother Cats” app. It includes the objectives, test responsibilities, entry and exit criteria, scope, schedule major milestones, entry and exit criteria and approach. This document has clearly identified what the test deliverables will be, and what is deemed in and out of scope.

**QUALITY OBJECTIVES**

A primary objective of testing is to: assure that the system meets the full requirements, including quality requirements (functional and non-functional requirements) and fit metrics for each quality requirement and satisfies the use case scenarios and maintain the quality of the product. At the end of the project development cycle, the user should find that the project has met or exceeded all of their expectations as detailed in the requirements.

Any changes, additions, or deletions to the requirements document, Functional Specification, or Design Specification will be documented and tested at the highest level of quality allowed within the remaining time of the project and within the ability of the test team.

**Secondary Objectives**

The secondary objectives of testing will be to: identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate matter before release. As an objective, this requires careful and methodical testing of the application to first ensure all areas of the system are scrutinized and, consequently, all issues (bugs) found are dealt with appropriately.

**TEST APPROACH**

The approach, that used, is Analytical therefore, in accordance to requirements-based strategy, where an analysis of the requirements specification forms the basis for planning,

estimating and designing tests. Test cases will be created during exploratory testing. All test types are determined in Test Strategy.

Team also must used experience-based testing and error guessing utilize testers' skills and intuition, along with their experience with similar applications or technologies.

The project is using an agile approach, with weekly iterations. At the end of each week the requirements identified for that iteration will be delivered to the team and will be tested.

**ROLES AND RESPONSIBILITIES**

| Role | Staff Member | Responsibilities |
| --- | --- | --- |
| Project  Manager |  | 1. Acts as a primary contact for development and QA team.  2. Responsible for Project schedule and the overall success of the project. |
| Developer Lead |  | 1. Participation in the project plan creation/update process. 2.Planning and organization of test process for the release. 3.Coordinate with QA analysts/engineers on any issues/problems encountered during testing.  4.Report progress on work assignments to the PM |
| QA    IOS Developer  Android developer |  | 1. Understand requirements  2. Writing and executing Test cases  3. Preparing RTM  4. Reviewing Test cases, RTM  5. Defect reporting and tracking  6. Retesting and regression testing  7. Bug Review meeting  8. Preparation of Test Data  9. Coordinate with Developer Lead for any issues or problems encountered during test preparation/execution/defect handling. |

**ENTRY AND EXIT CRITERIА**

**Entry:**

ξ All test hardware platforms must have been successfully installed, configured, and functioning properly.

ξ All the necessary documentation, design, and requirements information should be available that will allow testers to operate the system and judge the correct behavior. ξ All the standard software tools including the testing tools must have been successfully installed and functioning properly.

ξ Proper test data is available.

ξ The test environment such as, lab, hardware, software, and system administration support should be ready.

ξ QA resources have completely understood the requirements

ξ QA resources have sound knowledge of functionality

ξ Reviewed test scenarios, test cases and RTM

**Exit Criteria**

ξ A certain level of requirements coverage has been achieved.

ξ No high priority or severe bugs are left outstanding.

ξ All high-risk areas have been fully tested, with only minor residual risks left outstanding.

ξ Cost – when the budget has been spent.

ξ The schedule has been achieved

**TEST STRATEGY**

**Testing types**

Functional Testing:

Functional testing is carried out in order to find out unexpected behavior of the report. The characteristic of functional testing are to provide correctness, reliability, testability and accuracy of the report output/data.

System Testing:

System testing of software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

Performance Testing:

- Check the optimal time the photo is loaded

- Check the operation of the system under load

User acceptance testing:

The purpose behind user acceptance testing is to conform that system is developed according to the specified user requirements and is ready for operational use.

**Bug Severity and Priority Definition**

Bug Severity and Priority fields are both very important for categorizing bugs and prioritizing if and when the bugs will be fixed. The bug Severity and Priority levels will be defined as outlined in the following tables below. Testing will assign a severity level to all bugs. The Test Lead will be responsible to see that a correct severity level is assigned to each bug.

Development Lead and Project Manager will participate in bug review 10 

meetings to assign the priority of all currently active bugs. This meeting will be known as “Bug Triage Meetings”. The QA Lead is responsible for setting up these meetings on a routine basis to address the current set of new and existing but unresolved bugs.

**Severity List**

The tester entering a bug into GForge is also responsible for entering the bug Severity.

| **Severity ID Severity** |  | **Severity Description** |
| --- | --- | --- |
| 1 | Critical  **Level** | The module/product crashes or the bug causes non recoverable conditions. System crashes, GP Faults, or database or file corruption, or potential data loss, program hangs requiring reboot are all examples of a Sev. 1 bug. |
| 2 | High | Major system component unusable due to failure or incorrect functionality. Sev. 2 bugs cause serious problems such as a lack of functionality, or insufficient or unclear error messages that can have a major impact to the user, prevents other areas of the app from being tested, etc. Sev. 2 bugs can have a work around, but the work around is inconvenient or difficult. |
| 3 | Medium | Incorrect functionality of component or process. There is a simple work around for the bug if it is Sev. 3. |
| 4 | Minor | Documentation errors or signed off severity 3 bugs. |

Priority List

| **Priority** | **Priority Level** | **Priority Description** |
| --- | --- | --- |
| 1 | Must Fix | This bug must be fixed immediately; the product cannot ship with this bug. |
| 2 | Should Fix | These are important problems that should be fixed as soon as possible. It would be an embarrassment to the company if this bug shipped. |

| 3 | Fix When Have Time | The problem should be fixed within the time available. If the bug does not delay shipping date, then fix it. |
| --- | --- | --- |
| 4 | Low Priority | It is not important (at this time) that these bugs be addressed. Fix these bugs after all other bugs have been fixed. Enhancements/ Good to have features incorporated just are out of the current scope. |

**RESOURCE AND ENVIRONMENT NEEDS**

**Testing Tools**

| Process | Tool |
| --- | --- |
| Test case creation | Microsoft Excel |
| Test case tracking | Microsoft Excel |
| Test case execution | Manual, Selenium |
| Test case management | Microsoft Excel |
| Defect management | Microsoft Word |
| Test reporting | PDF |
| Check list creating | Microsoft Excel |
| Project structure | Mind Map (.xmind) |

**Test Environment**

ξ Support level 1 (browsers):

ξ Windows 10 and above: Chrome (latest), Firefox (latest), Safari (latest)

ξ Mac OS Big Sur and above: Chrome (latest), Firefox (latest), Safari (latest)

ξ IPhone with ios 14 and above, iPad 3,Samsung with Android 10 and above, Google Pixel with android 10 and above

ξ Support level 2:

ξ Windows 10: Chrome (latest), Firefox (latest), Safari (latest)