

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

09.03.04 - Программная инженерия
Профиль направления подготовки бакалавриата
“Системное и прикладное программное обеспечение”

ОТЧЕТ ПО ДИСЦИПЛИНЕ
«ВЕРИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

Выполнил:

студент 4 курса группы 22407

Э. А. Шараускас _____
подпись

Руководитель:

К. А. Кулаков, к.ф-м.н., доцент

подпись

Итоговая оценка

оценка

Содержание

1	Объект тестирования	3
1.1	Описание объекта тестирования	3
1.1.1	Модули и подсистемы	3
1.2	Функциональные возможности объекта тестирования	5
1.2.1	Тестируемые функции	6
2	Стратегия тестирования	7
2.1	Стратегия блочного тестирования	7
2.1.1	Критерии прохождения блочного тестирования	7
2.2	Стратегия интеграционного тестирования	7
2.2.1	Критерии прохождения интеграционного тестирования	8
2.3	Стратегия аттестационного тестирования	8
2.4	Стратегия выполнения специальных видов тестов	8
2.5	Условия начала, окончания и перехода между этапами тестирования	8
3	Детальный план тестирования	8
3.1	Блочное тестирование	8
3.2	Интеграционное тестирование	16
3.3	Аттестационное тестирование	18
3.4	Нагрузочное тестирование	21
4	Результаты тестирования	23
4.1	Блочное тестирование	23
4.2	Интеграционное тестирование	24
4.3	Аттестационное тестирование	24
4.4	Нагрузочное тестирование	25
4.5	Пример кода блочных тестов	25
4.6	Пример кода интеграционных тестов	26
4.7	Журнал найденных ошибок	27
4.8	Архитектура приложения	28
4.9	Покрываемость кода тестами	29
5	Заключение	31

1 Объект тестирования

1.1 Описание объекта тестирования

В связи со сложностью послушать музыку на своей устройстве, мне понравилась идея создать свое десктоп-приложение. Поэтому объектом тестирования является «RuShopify». Приложение предоставляет пользователю возможность воспроизведения аудио в форматах MP3, MP4, AAC, FLAC из локального хранилища устройства. Приложение реализовано на языке Python3.

Используемые библиотеки языка Python:

1. `os` – для работы с операционной системой;
2. `sys` – для доступа к некоторым переменным и функциям, взаимодействующим с интерпретатором python;
3. `time` – для работы со временем;
4. `random` – для генерации случайной последовательности воспроизведения.
5. `PyQt5` – для описания архитектуры форм приложения при помощи Qt-виджетов библиотеки.

1.1.1 Модули и подсистемы

1. `class MainWindow` – модуль компоновки и отрисовки приложения. Наследован от `QtWidgets.QWidget`, содержит в себе описание интерфейса приложения с использованием модулей и функций библиотеки `PyQt5`:
 - (a) Имеет единственный подмодуль, содержащий метод `show()` – показывает скомпонованный интерфейс.
2. `class MediaLibrary` – модуль взаимодействия приложения с файловой системой и работы с добавленной медиатекой. Наследован от `QtWidgets.QListWidget`, содержит в себе функции чтения из файловой системы и работы с медиатекой, представленной в виде списка;
 - (a) `def hide_show_ml(self, hide):` – подмодуль, реализующий функцию отображения или скрытия медиатеки. Где `hide` – bool-овое значение отображения;

- (b) `def open_mp3_file(self)`: – подмодуль, реализующий функцию чтения файлов аудиоформатов MP3, MP4, AAC, FLAC и добавления их в медиатеку приложения;
- (c) `def make_item(self, file)`: – подмодуль, реализующий функцию создания одного элемента медиатеки, вызывается из `open_mp3_file` для каждого файла аудио формата. Где `file` – путь к аудиофайлу в файловой системе. В себе содержит кнопку, для перемещения аудио в плейлист и кнопку удаления из медиатеки;
- (d) `def filter_ml(self, property, value)`: – подмодуль, реализующий функцию фильтрации медиатеки. Где `property` – свойство фильтрации, `value` – значение свойства фильтрации;
- (e) `def add_item_in_pl(self, item)`: – подмодуль, реализующий функцию добавления аудиозаписи в плейлист воспроизведения, вызывается из `make_item`. Где `item` – строка из которой поступил сигнал на добавление, содержит в себе путь к аудиофайлу в файловой системе;
- (f) `def add_all_items_in_pl(self)`: – подмодуль, реализующий функцию добавления всей медиатеки в плейлист воспроизведения;
- (g) `def delete_item(self, item)`: – подмодуль, реализующий функцию удаления аудиозаписи из медиатеки, вызывается из `make_item`. Где `item` – строка из которой поступил сигнал на удаление;
- (h) `def delete_all_items(self)`: – подмодуль, реализующий функцию очищения медиатеки;

3. `class Playlist` – модуль для управления плейлистом воспроизведения. Наследован от `QtWidgets.QListWidget`, содержит в себе функции работы с плейлистом воспроизведения, представленного в виде списка;

- (a) `def make_item_with_audio(self, audio)`: – подмодуль, реализующий функцию создания одного элемента плейлиста, вызывается из `add_item_in_pl`. Где `audio` – путь к аудиофайлу в файловой системе. В себе содержит кнопку удаления из плейлиста;
- (b) `def change_play_mode(self, playmode)`: – подмодуль, реализующий функцию изменения режима воспроизведения. Где `playmode` – int-овое значение;
- (c) `def delete_item_with_audio(self, item)`: – подмодуль, реализующий функцию

удаления аудиозаписи из плейлиста, вызывается из `make_item_with_audio`. Где `item` – строка из которой поступил сигнал на удаление;

(d) `def delete_all_items_with_audio(self)`: – подмодуль, реализующий функцию очищения плейлиста;

4. `class AudioInPlay()`: – модуль для управления воспроизводимой аудиозаписью. Содержит в себе интерфейсы с информацией об аудиозаписи и функции для взаимодействия пользователя с ней.

(a) `def load_audio(self, audio)`: – подмодуль, реализующий функцию заполнения информации об воспроизводимой аудиозаписи. Где `audio` – путь к аудиофайлу в файловой системе;

(b) `def change_play_pause(self, pause)`: – подмодуль, реализующий функцию остановки и воспроизведения аудиозаписи. Где `pause` – bool-овое значение паузы;

(c) `def position_changed(self, position)`: – подмодуль, реализующий функцию установки временной позиции в аудиозаписи, вызывается из `load_audio`. Где `position` – позиция во времени воспроизводимой аудиозаписи;

(d) `def duration_changed(self, duration)`: – подмодуль, реализующий функцию установки длительности аудиозаписи, вызывается из `load_audio`. Где `duration` – длительность аудиозаписи;

(e) `def playlist_next(self)`: – подмодуль, реализующий функцию переключения на следующую аудиозапись;

(f) `def playlist_prev(self)`: – подмодуль, реализующий функцию переключения на предыдущую аудиозапись;

(g) `def set_position(self, position)`: – подмодуль, реализующий функцию перемотки аудиозаписи. Где `position` – позиция во времени воспроизводимой аудиозаписи.

1.2 Функциональные возможности объекта тестирования

1. Работа с медиатекой приложения:

(a) Открытие области медиатеки;

(b) Скрытие области медиатеки;

(c) Добавление аудио файлов из локального хранилища в медиатеку приложения;

- (d) Просмотр медиатеки;
- (e) Поиск аудио файла по подстроке;
- (f) Сортировка медиатеки по имени;
- (g) Фильтрация медиатеки по исполнителю;
- (h) Фильтрация медиатеки по альбому;
- (i) Фильтрация медиатеки по жанру;
- (j) Удаление аудио файлов из медиатеки.

2. Работа с плейлистом воспроизведения:

- (a) Добавление аудиозаписей из медиатеки в плейлист воспроизведения;
- (b) Просмотр плейлиста;
- (c) Изменение режима воспроизведения плейлиста (циклическое воспроизведение плейлиста, циклическое воспроизведение аудиозаписи, случайное воспроизведение плейлиста);
- (d) Изменение порядка аудиозаписей в плейлисте;
- (e) Удаление аудиозаписей из плейлиста.

3. Работа с воспроизводимой аудиозаписью:

- (a) Воспроизведение аудиозаписи из плейлиста;
- (b) Просмотр информации об аудиозаписи;
- (c) Установка паузы воспроизводимой аудиозаписи;
- (d) Воспроизведение аудиозаписи после установки паузы;
- (e) Переключение на следующую аудиозапись из плейлиста;
- (f) Переключение на предыдущую аудиозапись из плейлиста;
- (g) Изменение позиции воспроизводимой аудиозаписи (перемотка).

1.2.1 Тестируемые функции

Блочное тестирование проводится для всех описанных функций, которые принимают входные аргументы, так как они напрямую влияют на работу пользователя с приложением. Тестируются функции описанные в подмодулях в пункте 1.1.1.

Интеграционное тестирование проводится для следующих связей:

1. Функция `make_item_with_audio` – модуль `MediaLibrary` и модуль `PlayList`;
2. Функции `playlist_prev` и `playlist_next` – модуль `PlayList` и модуль `AudioInPlay`;

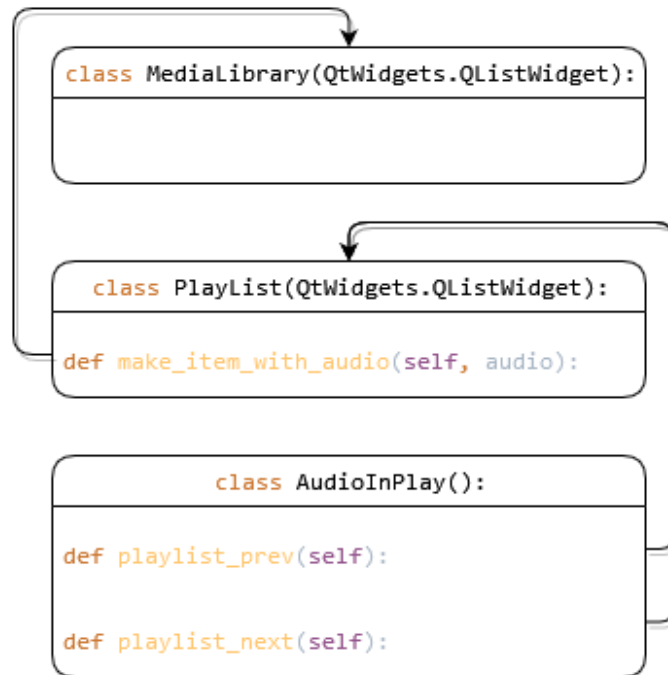


Рис. 1: Связи модулей

2 Стратегия тестирования

2.1 Стратегия блочного тестирования

Блочное тестирование производится при помощи фреймворка `PyTest`. Код блочных тестов хранится рядом с кодом модулей, название файлов должно начинаться с «`test_`» и содержать имя тестируемого модуля.

2.1.1 Критерии прохождения блочного тестирования

1. Тест считается пройденным, если все его кейсы завершились успешно;
2. Тестирование считается пройденным если все тесты пройдены.

2.2 Стратегия интеграционного тестирования

Интеграционное тестирование также проводится при помощи фреймворка `PyTest`. Код интеграционных тестов хранится рядом с кодом модулей, название файлов должно

начинаться с «test_» и содержать имена взаимодействующих модулей. Отчеты о покрытии для интеграционного тестирования создаются отдельно.

Используется интеграция "сверху вниз заглушки низкоуровневых модулей реализованы в файлах с кодом тестов.

2.2.1 Критерии прохождения интеграционного тестирования

1. Тест считается пройденным, если все его кейсы завершились успешно;
 - (а) Интеграционным тест считается не пройденным, если хотя бы один блочный тест низкоуровневых модулей не пройден.
2. Тестирование считается пройденным если все тесты пройдены.

2.3 Стратегия аттестационного тестирования

Аттестационное тестирование интерфейса приложения проводится в ручном режиме на устройстве с операционной системой Windows 10.

2.4 Стратегия выполнения специальных видов тестов

2.5 Условия начала, окончания и перехода между этапами тестирования

Блочное тестирование выполняется автоматически при любом изменении в репозитории проекта. Блочное тестирование продолжает выполняться, если какой-либо из тестов не пройден.

Интеграционное тестирование модуля проводится после прохождения блочного для модулей более низкого уровня.

Аттестационное тестирование проводится при окончании разработки очередной функции системы и прохождения интеграционного тестирования. Также аттестационное тестирование проводится перед окончанием разработки проекта.

3 Детальный план тестирования

3.1 Блочное тестирование

ID теста	Б1 (test_hideShowMl_hide)
Объект тестирования	class MediaLibrary
Цель тестирования	Проверить обработку нажатия на кнопку свертки медиатеки
Входные данные	Вызов метода hide_show_ml с параметром hide=true
Ожидаемый результат	<i>hide_show_ml.setVisibleMediaLibrary false</i>
Тип	Позитивный

ID теста	Б2 (test_hideShowMl_show)
Объект тестирования	class MediaLibrary
Цель тестирования	Проверить обработку нажатия на кнопку раскрытия медиатеки
Входные данные	Вызов метода hide_show_ml с параметром hide=false
Ожидаемый результат	hide_show_ml изменяет значение свойства setVisible модуля MediaLibrary на true
Тип	Позитивный

ID теста	Б3 (test_hideShowMl_show_n)
Объект тестирования	class MediaLibrary
Цель тестирования	Проверить обработку повторного нажатия на кнопку раскрытия медиатеки
Входные данные	Вызов метода hide_show_ml с параметром hide=false
Ожидаемый результат	hide_show_ml не изменяет значение свойства setVisible модуля MediaLibrary
Тип	Негативный

ID теста	Б4 (test_makeItem_new)
Объект тестирования	class MediaLibrary
Цель тестирования	Проверить добавление уникальной аудиозаписи в медиатеку

Входные данные	Вызов метода <code>make_item</code> с параметром <code>file="C:/Users/user/Downloads/Шатунов - Белые Розы"</code>
Ожидаемый результат	<code>make_item</code> создает объект <code>QtWidgets.QListWidgetItem</code> в модуле <code>MediaLibrary</code> с дочерним подмодулем <code>QLabel</code> , свойство которого <code>text="C:/Users/user/Downloads/Шатунов - Белые Розы"</code> и подмодулем <code>QPushButton</code> с именем <code>item_delete_button</code>
Тип	Позитивный

ID теста	Б5 (<code>test_makeItem_duplicate</code>)
Объект тестирования	<code>class MediaLibrary</code>
Цель тестирования	Проверить добавление повторяющейся аудиозаписи в медиатеку
Входные данные	Повторный вызов метода <code>make_item</code> с параметром <code>file="C:/Users/user/Downloads/Шатунов - Белые Розы"</code>
Ожидаемый результат	<code>make_item</code> создает объект <code>QtWidgets.QListWidgetItem</code> в модуле <code>MediaLibrary</code> с дочерним подмодулем <code>QLabel</code> , свойство которого <code>text="C:/Users/user/Downloads/Шатунов - Белые Розы"</code> и подмодулем <code>QPushButton</code> с именем <code>item_delete_button</code> , дублирующий уже имеющийся в медиатеке, но с другим индексом
Тип	Негативный

ID теста	Б6 (<code>test_filter_ml_author</code>)
Объект тестирования	<code>class MediaLibrary</code>
Цель тестирования	Проверить метод вывода списка исполнителей
Входные данные	Вызов метода <code>filter_ml</code> с параметрами <code>property="author value=null"</code>

Ожидаемый результат	filter_ml формирует названия исполнителей медиатеки в массив данных List, и заполняет модуль MediaLibrary данными из List. В модуле MediaLibrary остается лишь список исполнителей из медиатеки
Тип	Позитивный

ID теста	Б7 (test_filter_ml_author_value)
Объект тестирования	class MediaLibrary
Цель тестирования	Проверить метод вывода списка аудиозаписей одного исполнителя
Входные данные	Вызов метода filter_ml с параметрами property="author value="Шатунов"
Ожидаемый результат	filter_ml выполняет поиск подмодулей типа QtWidgets.QListWidgetItem модуля MediaLibrary, из item-ов, у которых свойство author содержит подстроку "Шатунов" формирует список аудиозаписей в массив данных List, и заполняет модуль MediaLibrary данными из List. В модуле MediaLibrary остается лишь список аудиозаписей исполнителя
Тип	Позитивный

ID теста	Б8 (test_filter_ml_author_err)
Объект тестирования	class MediaLibrary
Цель тестирования	Проверить метод вывода списка аудиозаписей несуществующего исполнителя
Входные данные	Вызов метода filter_ml с параметрами property="author value="abc"

Ожидаемый результат	filter_ml выполняет поиск подмодулей типа QtWidgets.QListWidgetItem модуля MediaLibrary, из item-ов, у которых свойство author содержит подстроку "abc" формирует список аудиозаписей в массив данных List, и заполняет модуль MediaLibrary данными из List. Формируется пустой List, соответственно MediaLibrary становится пустой
Тип	Негативный

ID теста	Б9 (test_filter_ml_find)
Объект тестирования	class MediaLibrary
Цель тестирования	Проверить метод вывода списка аудиозаписей, в свойствах name, author, genre которого встречается подстрока из поиска
Входные данные	Вызов метода filter_ml с параметрами property=null, value="pop"
Ожидаемый результат	filter_ml выполняет поиск подмодулей типа QtWidgets.QListWidgetItem модуля MediaLibrary, из item-ов, у которых свойство name, author или genre содержит подстроку "pop" формирует список аудиозаписей в массив данных List, и заполняет модуль MediaLibrary данными из List
Тип	Позитивный

ID теста	Б10 (test_filter_ml_find_err)
Объект тестирования	class MediaLibrary
Цель тестирования	Проверить метод вывода списка аудиозаписей при некорректном значении property
Входные данные	Вызов метода filter_ml с параметрами property="asdaasd value=null

Ожидаемый результат	Ошибка отсутствия свойства "abc" объекта, путь к которому находится в подмодуле QtWidgets.QListWidgetItem модуля MediaLibrary
Тип	Негативный

ID теста	Б11 (test_deleteItem_inMediaLibrary)
Объект тестирования	class MediaLibrary
Цель тестирования	Проверить удаление строки медиатеки
Входные данные	Вызов метода delete_item с параметром item=1
Ожидаемый результат	delete_item удаляет объект QtWidgets.QListWidgetItem в модуле MediaLibrary значение свойства которого row=1
Тип	Позитивный

ID теста	Б12 (test_deleteItem_inMediaLibrary_out)
Объект тестирования	class MediaLibrary
Цель тестирования	Проверить удаление строки медиатеки за ее границами
Входные данные	Вызов метода delete_item с параметром item=99
Ожидаемый результат	Возврат ошибки outOfRange <QListWidget>
Тип	Негативный

ID теста	Б13 (test_changePlayMode_random)
Объект тестирования	class PlayList
Цель тестирования	Проверить метод изменения режима воспроизведения плейлиста на случайный
Входные данные	Вызов метода change_play_mode с параметром playmode=0
Ожидаемый результат	change_play_mode изменяет значение свойства playmode=1 модуля PlayList
Тип	Позитивный

ID теста	Б14 (test_changePlayMode_cycle_one)
Объект тестирования	class PlayList
Цель тестирования	Проверить метод изменения режима воспроизведения плейлиста на циклическое воспроизведение аудиозаписи
Входные данные	Вызов метода change_play_mode с параметром playmode=1
Ожидаемый результат	change_play_mode изменяет значение свойства playmode=2 модуля PlayList
Тип	Позитивный

ID теста	Б15 (test_changePlayMode_cycle)
Объект тестирования	class PlayList
Цель тестирования	Проверить метод изменения режима воспроизведения плейлиста на циклическое воспроизведение плейлиста
Входные данные	Вызов метода change_play_mode с параметром playmode=2
Ожидаемый результат	change_play_mode изменяет значение свойства playmode=0 модуля PlayList
Тип	Позитивный

ID теста	Б16 (test_changePlayMode_)
Объект тестирования	class PlayList
Цель тестирования	Проверить метод изменения режима воспроизведения плейлиста с недопустимым параметром playmode
Входные данные	Вызов метода change_play_mode с параметром playmode=10
Ожидаемый результат	change_play_mode не изменяет значение свойства playmode модуля PlayList
Тип	Негативный

ID теста	Б17 (test_deleteItemWithAudio_inPlayList)
Объект тестирования	class PlayList
Цель тестирования	Проверить удаление строки плейлиста
Входные данные	Вызов метода delete_item_with_audio с параметром item=1
Ожидаемый результат	delete_item_with_audio удаляет объект QtWidgets.QListWidgetItem в модуле Playlist значение свойства которого row=1
Тип	Позитивный

ID теста	Б18 (test_deleteItemWithAudio_inPlayList_out)
Объект тестирования	class PlayList
Цель тестирования	Проверить удаление строки плейлиста за его границами
Входные данные	Вызов метода delete_item_with_audio с параметром item=99
Ожидаемый результат	Возврат ошибки outOfRange <QListWidget>
Тип	Негативный

ID теста	Б19 (test_change_play_pause_pause)
Объект тестирования	class AudioInPlay
Цель тестирования	Проверить остановку воспроизведения аудиозаписи
Входные данные	Вызов метода change_play_pause с параметром pause=true
Ожидаемый результат	change_play_pause изменяет значение свойства pause=true модуля AudioInPlay
Тип	Позитивный

ID теста	Б20 (test_change_play_pause_play)
Объект тестирования	class AudioInPlay
Цель тестирования	Проверить воспроизведение аудиозаписи после остановки

Входные данные	Вызов метода <code>change_play_pause</code> с параметром <code>pause=false</code>
Ожидаемый результат	<code>change_play_pause</code> изменяет значение свойства <code>pause=false</code> модуля <code>AudioInPlay</code>
Тип	Позитивный

ID теста	B21 (<code>test_position_changed</code>)
Объект тестирования	<code>class AudioInPlay</code>
Цель тестирования	Проверить изменение позиции аудиозаписи после ее загрузки в модуль <code>AudioInPlay</code>
Входные данные	Вызов метода <code>position_changed</code> с параметром <code>position=0</code>
Ожидаемый результат	<code>position_changed</code> изменяет значение свойства <code>position=0</code> модуля <code>AudioInPlay</code>
Тип	Позитивный

ID теста	B22 (<code>test_duration_changed</code>)
Объект тестирования	<code>class AudioInPlay</code>
Цель тестирования	Проверить изменение длительности аудиозаписи после ее загрузки в модуль <code>AudioInPlay</code>
Входные данные	Вызов метода <code>duration_changed</code> с параметром <code>position=0</code>
Ожидаемый результат	<code>duration_changed</code> изменяет значение свойства <code>duration=значению свойства duration</code> объекта из параметра <code>audio</code> модуля <code>AudioInPlay</code>
Тип	Позитивный

3.2 Интеграционное тестирование

ID теста	И1 (<code>test_mediaLibrary_playList_add</code>)
Объект тестирования	Связь модулей <code>class MediaLibrary</code> и <code>class PlayList</code>

Цель тестирования	Проверить интеграцию модулей медиатеки и плейлиста
Входные данные	Вызов метода <code>make_item_with_audio</code> модуля <code>PlayList</code> из модуля <code>MediaLibrary</code> с параметром <code>audio=C:/Users/user/Downloads/Шатунов - Белые Розы'</code>
Ожидаемый результат	<code>make_item_with_audio</code> создает объект <code>QtWidgets.QListWidgetItem</code> в модуле <code>PlayList</code> с дочерним подмодулем <code>QLabel</code> , свойство которого <code>text=C:/Users/user/Downloads/Шатунов - Белые Розы"</code> и подмодулем <code>QPushButton</code> с именем <code>item_delete_button</code>
Тип	Позитивный

ID теста	И2 (<code>test_mediaLibrary_playList_add_err</code>)
Объект тестирования	Связь модулей <code>class MediaLibrary</code> и <code>class PlayList</code>
Цель тестирования	Проверить интеграцию модулей медиатеки и плейлиста при передаче некорректных данных
Входные данные	Вызов метода <code>make_item_with_audio</code> модуля <code>PlayList</code> из модуля <code>MediaLibrary</code> с параметром <code>audio=</code>
Ожидаемый результат	<code>make_item_with_audio</code> создает объект <code>QtWidgets.QListWidgetItem</code> в модуле <code>PlayList</code> с дочерним подмодулем <code>QLabel</code> , свойство которого <code>text=</code> и подмодулем <code>QPushButton</code> с именем <code>item_delete_button</code>
Тип	Негативный

ID теста	И3 (<code>test_playList_audioInPlay_load</code>)
Объект тестирования	Связь модулей <code>class PlayList</code> и <code>class AudioInPlay</code>
Цель тестирования	Проверить интеграцию модулей плейлиста и воспроизводимой аудиозаписи

Входные данные	Вызов метода load_audio модуля AudioInPlay из модуля Playlist с параметром audio="C:/Users/user/Downloads/Шатунов - Белые Розы'
Ожидаемый результат	load_audio изменяет значение свойства text="C:/Users/user/Downloads/Шатунов - Белые Розы' QtWidgets.QLabel в модуле AudioInPlay. Вызывает методы: position_changed с параметром position=0 и duration_changed с параметром duration=значению свойства duration объекта из параметра audio модуля AudioInPlay
Тип	Позитивный

3.3 Аттестационное тестирование

ID теста	A1
Описание	Запуск приложения
Цель тестирования	Проверка запуска приложения
Ход тестирования	1. Открытие исполняемого файла приложения
Ожидаемый результат	Приложение запустилось

ID теста	A2
Описание	Пополнение медиатеки
Цель тестирования	Проверка пополнения медиатеки из локального хранилища устройства

Ход тестирования	<ol style="list-style-type: none"> 1. Нажатие на кнопку "AddMedia" 2. Выбор файлов в локальном хранилище 3. Нажатие кнопки "Open" в диалоговом окне
Ожидаемый результат	Медиатека пополнилась путями на файлы в локальном хранилище

ID теста	A3
Описание	Очистка медиатеки
Цель тестирования	Проверка очищения медиатеки
Ход тестирования	<ol style="list-style-type: none"> 1. Нажатие на кнопку "CleanMediaList"
Ожидаемый результат	Медиатека очистилась

ID теста	A4
Описание	Фильтрация медиатеки по исполнителю Шатунов
Цель тестирования	Проверка фильтрации медиатеки
Ход тестирования	<ol style="list-style-type: none"> 1. Нажатие на кнопку "AuthorFilter" 2. В медиатеке отображается список исполнителей медиатеки 3. Нажатие на элемент списка с текстовым значением Шатунов"
Ожидаемый результат	Получен и помещен в медиатеку список аудиозаписей автором которых указан "Шатунов"

ID теста	A5
Описание	Пополнение плейлиста из медиатеки
Цель тестирования	Проверка пополнения плейлиста аудиозаписями из медиатеки
Ход тестирования	1. Нажатие на кнопку "AddMediaInPlaylist"
Ожидаемый результат	Плейлист пополнился аудиозаписями из медиатеки

ID теста	A6
Описание	Остановка аудиозаписи
Цель тестирования	Проверка паузы воспроизведения
Ход тестирования	1. Нажатие на кнопку "PausePlay"
Ожидаемый результат	Воспроизведение аудиозаписи остановилось, позиция воспроизведения зафиксировалась

ID теста	A7
Описание	Воспроизведение после остановки аудиозаписи
Цель тестирования	Проверка паузы воспроизведения
Ход тестирования	1. Нажатие на кнопку "PausePlay"
Ожидаемый результат	Воспроизведение аудиозаписи продолжилось, позиция воспроизведения изменяется

ID теста	A8
Описание	Изменение режима воспроизведения плейлиста
Цель тестирования	Проверка смены режима воспроизведения

Ход тестирования	1. Нажатие на кнопку "PlayMode"
Ожидаемый результат	Изменяется иконка на кнопке "PlayMode"

ID теста	A9
Описание	Удаление аудиозаписи из плейлиста
Цель тестирования	Проверка удаления аудиозаписи из плейлиста
Ход тестирования	1. Нажатие на кнопку "ItemDeleteButton"
Ожидаемый результат	Из плейлиста удаляется строка с аудиозаписью, напротив которой была нажата кнопка

3.4 Нагрузочное тестирование

ID теста	H1
Объект тестирования	class MediaLibrary
Цель тестирования	Проверка работы функции make_item
Входные данные	Список из 200 путей на локальные файлы
Ожидаемый результат	Время выполнения функции не превышает 1 секунды. Приложение продолжает функционировать. Каждый элемент медиатеки доступен при ее прокрутке

ID теста	H2
Объект тестирования	class MediaLibrary
Цель тестирования	Проверка работы функции make_item
Входные данные	Список из 1000 путей на локальные файлы

Ожидаемый результат	Время выполнения функции не превышает 2 секунд. Приложение продолжает функционировать. Каждый элемент медиатеки доступен при ее прокрутке
----------------------------	--

ID теста	НЗ
Объект тестирования	class MediaLibrary
Цель тестирования	Проверка работы функции make_item
Входные данные	Список из 10000 путей на локальные файлы
Ожидаемый результат	Время выполнения функции не превышает 10 секунд. Приложение продолжает функционировать. Каждый элемент медиатеки доступен при ее прокрутке

4 Результаты тестирования

4.1 Блочное тестирование

ID теста	Дата	Результат	Фактический результат	Отчёт
Б1	15.12.23	Пройден		
Б2	15.12.23	Пройден		
Б3	15.12.23	Пройден		
Б4	15.12.23	Пройден		
Б5	15.12.23	Пройден		
Б6	15.12.23	Пройден		
Б7	15.12.23	Пройден		
Б8	15.12.23	Пройден		
Б9	15.12.23	Пройден		
Б10	15.12.23	Пройден		
Б11	15.12.23	Пройден		
Б12	15.12.23	Не пройден	Метод delete_item завершился с ошибкой выхода за границы массива	Ошибка 1: Отсутствует проверка выхода за границы списка
Б13	15.12.23	Пройден		
Б14	15.12.23	Пройден		
Б15	15.12.23	Пройден		
Б16	15.12.23	Пройден		
Б17	15.12.23	Пройден		
Б18	15.12.23	Не пройден	Метод delete_item_with_audio завершился с ошибкой выхода за границы массива	Ошибка 2: Отсутствует проверка выхода за границы списка
Б19	15.12.23	Пройден		

Б20	15.12.23	Пройден		
Б21	15.12.23	Пройден		
Б22	15.12.23	Пройден		

4.2 Интеграционное тестирование

ID теста	Дата	Результат	Фактический результат	Отчёт
И1	15.12.23	Пройден		
И2	15.12.23	Пройден		
И3	15.12.23	Пройден		

4.3 Аттестационное тестирование

ID теста	Дата	Результат	Фактический результат	Отчёт
A1	15.12.23	Пройден		
A2	15.12.23	Пройден		
A3	15.12.23	Пройден		
A4	15.12.23	Пройден		
A5	15.12.23	Пройден		
A6	15.12.23	Пройден		
A7	15.12.23	Пройден		
A8	15.12.23	Пройден		
A9	15.12.23	Пройден		

4.4 Нагрузочное тестирование

ID теста	Дата	Результат	Фактический результат	Отчёт
H1	15.12.23	Пройден		
H2	15.12.23	Пройден		
H3	15.12.23	Пройден		

4.5 Пример кода блочных тестов

```
from PyQt5 import QtCore, QtGui, QtWidgets, \
    QtMultimedia
from PyQt5.QtCore import QUrl, QUrlQuery
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.QtMultimedia import QMediaContent, \
    QMediaPlayer, QMediaPlaylist

import random
import sys
import os
import time

from ru_shopify import PlayList
from ru_shopify import MediaLibrary
from ru_shopify import AudioInPlay

app = QtWidgets.QApplication([])

def test_hideShowMl_hide():
    hide = True

    media_library = MediaLibrary()
```

```
media_library.setVisible(not hide)

assert not media_library.visible
```

4.6 Пример кода интеграционных тестов

```
from PyQt5 import QtCore, QtGui, QtWidgets, \
    QtMultimedia
from PyQt5.QtCore import QUrl, QUrlQuery
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.QtMultimedia import QMediaContent, \
    QMediaPlayer, QMediaPlaylist

import random
import sys
import os
import time

from ru_shopify import PlayList
from ru_shopify import MediaLibrary
from ru_shopify import AudioInPlay

app = QtWidgets.QApplication([])

def test_playback_start():
    playlist = QMediaPlaylist()
    content = QMediaContent(QUrl.fromLocalFile("C:/Users/user/Downloads/s
    player = QMediaPlayer()

    playlist.addMedia(content)
```

```
player.setPlaylist(playlist)
```

```
player.play()
```

```
assert player.state() == QMediaPlayer.PlayingState
```

```
time.sleep(3)
```

```
assert player.duration() > 0
```

```
assert player.currentMedia().canonicalUrl() == content.canonicalUrl()
```

4.7 Журнал найденных ошибок

№ отчета об ошибке	1
Дата составления отчета	15.12.23
ID теста	Б12
Объект тестирования	class PlayList
Ожидаемый результат	Возврат ошибки <code>outOfRange <QListWidget></code>
Фактический результат	Возврат ошибки <code>outOfRange <QListWidget></code>
Воспроизводимость	Всегда
Приоритет	Некритическая ошибка
Вариант решения	Добавить проверку выхода за границы списка
Статус	Решено

4.8 Архитектура приложения

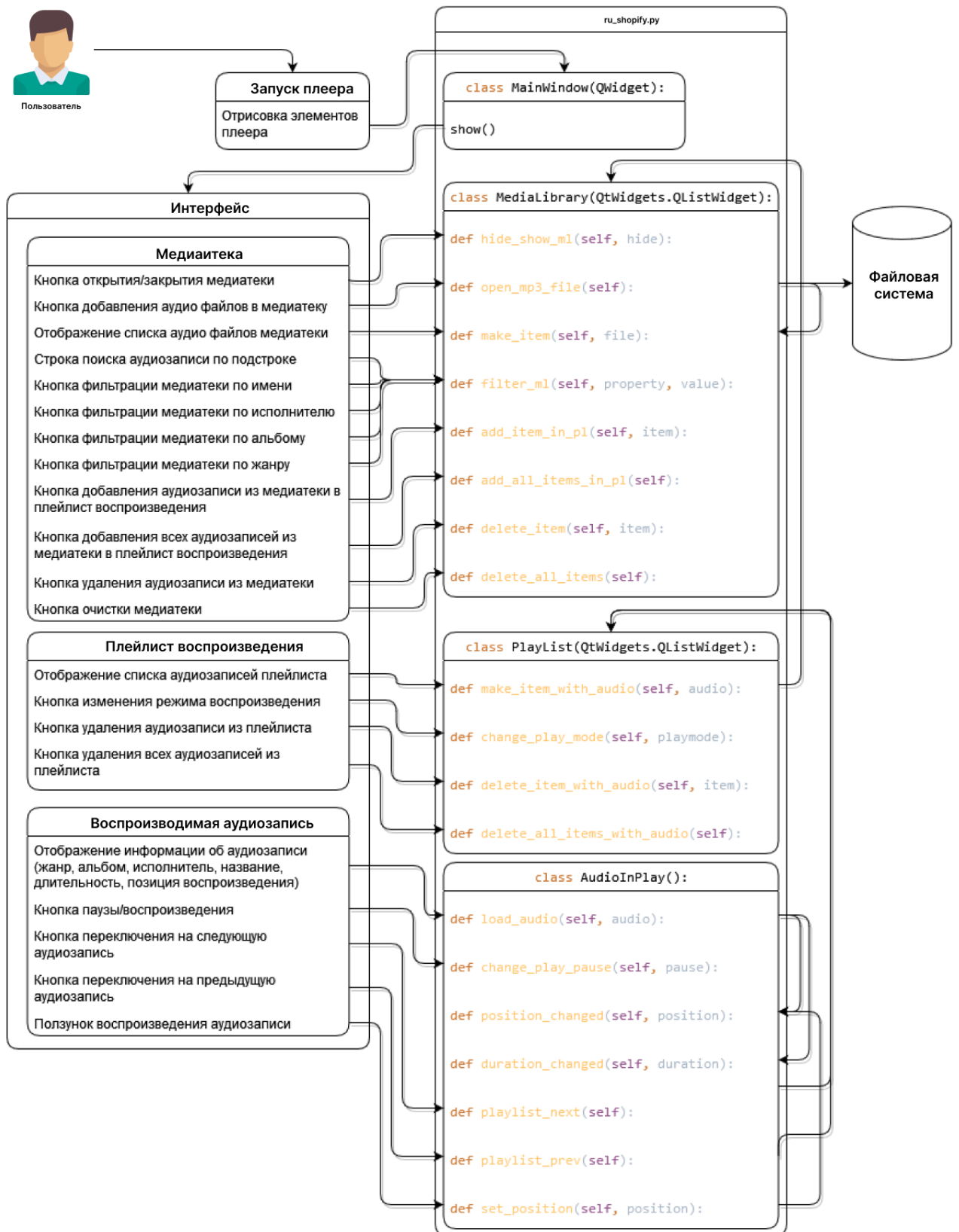


Рис. 2: Архитектура приложения

4.9 Покрытие кода тестами

Покрытие тестами модулей высчитывается при помощи `pytest-cov`.

```
test_medialibrary.py ✓✓✓
test_playlist.py    ✓✓✓
test_audioinplay.py ✓✓✓

----- coverage: platform win32, python 3.10.7-final-0 -----
Name                Stmts  Miss  Cover
-----
ru_shopify.py       887    178    80%
test_medialibrary.py 234     0   100%
test_playlist.py    135     0   100%
test_audioinplay.py  98     0   100%
-----
TOTAL                1354   178    87%
```

Рис. 3: Покрытие тестами

№ отчета об ошибке	2
Дата составления отчета	15.12.23
ID теста	Б18
Объект тестирования	class PlayList
Ожидаемый результат	Возврат ошибки outOfRange <QListWidget>
Фактический результат	Возврат ошибки outOfRange <QListWidget>
Воспроизводимость	Всегда
Приоритет	Некритическая ошибка
Вариант решения	Добавить проверку выхода за границы списка
Статус	Решено

5 Заключение

Проведено блочное, интеграционное, аттестационное тестирование десктоп-приложения «RuShopify».

В ходе тестирования модулей приложения было выявлено 2 не критических ошибки.

Ошибки №1 – 2 были исправлены. Предположительно, работа системы с технической точки зрения является работоспособной, все заявленные функции выполняются без ошибок.

Составлен отчёт о прохождении тестирования.