МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ

по учебной практике

Тема: UI-тестирование приложения ClickUp

Студент гр. 2303 _____ Степанов Д.А
Студент гр. 2300 _____ Войнов А.Н
Студент гр. 2303 _____ Репкин В.С.
Руководитель _____ Шевелева А.М.

Санкт-Петербург
2024

ЗАДАНИЕ

НА УЧЕБНУЮ ПРАКТИКУ

Студент Степанов Д.А. группа 2303

Студент Репкин В.С группа 2303

Студент Войнов А.Н группа 2300

Тема практики: UI-тестирование

Задание на практику:

Командное UI-тестирование веб-приложения ClickUp

Сроки прохождения практики: 25.06.2024 – 09.07.2024

Дата сдачи отчета: 06.07.2024

Дата защиты отчета: 09.07.2024

Студент _____ Степанов Д.А.

Студент _____ Репкин В.С.

Студент _____ Войнов А.Н

Руководитель _____ Шевелева А.М.

АННОТАЦИЯ

Целью практики является освоение фрейморков для автоматизации тестирования на языке Java: Selenide и Junit. А затем написании проекта по тестированию сайта clickup.com.

СОДЕРЖАНИЕ

	Введение	5
1.	Описание классов и методов	6
1.1.	Описание	6
1.1.1	Elements	6
1.1.2	Pages	11
1.1.3	Tests	26
1.1.4	Utils	29
1.2.	UML	29
2.	Описание тестов	31
2.1.	Тестирование создания и удаления задач, проверка различных функций	31
2.2.	Тестирование создания и удаления рабочих пространств с разными параметрами	36
2.3	Тестирование создания, удаления и перемещения документов.	42
	Заключение	48
	Список использованных источников	49
	Приложение А. Чеклист тестов	50
	Приложение Б. UML-диаграмма	59

ВВЕДЕНИЕ

Главной целью практики было тестирование пользовательского интерфейса (UI) сайта ClickUp, на языке Java. Для этого был изучен фреймворк для автоматизированного тестирования веб-приложений Selenide[1], фреймворк для автоматического Unit тестирования JUnit[2], также был изучен Maven для сборки проекта на языке Java. Реализованы 3 блока тестов, связанных с добавлением и удалением задач, созданием и удалением документов, созданием и удалением рабочих пространств.

1. ОПИСАНИЕ КЛАССОВ И МЕТОДОВ

1.1. Описание

Код проекта был разбит на 4 пакета: elements, pages, tests и utls.

Исходный код проекта можно посмотреть на гитхабе[3].

1.1.1. Elements

Пакет содержит классы, описывающие html-элементы страницы:

BaseElement - абстрактный базовый класс для представления веб-элементов. Этот класс предоставляет методы для работы с веб-элементами с использованием Selenide. Он содержит protected static final строки-заготовки, используемые для получения элементов с страницы методами Selenide. Строки формата "//%s[@id='%s']", куда впоследствии в конструкторе класса подставляются тип веб-элемента и значение его id или другого параметра в зависимости от строки. Также содержит поле baseElement типа SelenideElement, хранящее сам веб-элемент и строку xPath этого элемента. Методы, реализованные в BaseElement:

- protected BaseElement(String fullXpath) - конструктор, принимает xPath элемента и создает baseElement по нему.
- protected BaseElement(String xpath, String attributeValue, String elementType) - конструктор, принимает строку-заготовку, а также тип элемента и аргумент.
- protected BaseElement(String xpath, String attributeValue, String elementType, int index) - конструктор, используется в случае, когда на странице несколько элементов с одинаковым относительным xPath и надо получить один из них по индексу.
- public String getAttributeValue(String attributeName) - возвращает значение атрибута по его имени.
- public boolean isDisplayed() - проверяет отображается ли элемент на странице.

- public boolean isExists() - проверяет существует ли элемент на странице.
- public String getText() - получает текст из элемента.

Пакет Buttons - содержит классы для создания элемента ‘Кнопки’ с помощью различных XPATH

Класс Button - предназначен для представления кнопки на веб-странице, также предоставляет методы для взаимодействия с ней.

Конструкторы:

- private Button(String xPath, String param) - конструктор инициализирует кнопку с заданным xPath и значением атрибута
- protected Button(String xPath, String param, String htmlName) - конструктор инициализирует кнопку с заданным xPath, значением атрибута и именем HTML - элемента
- protected Button(String xPath) - конструктор инициализирует кнопку с заданным xPath
- protected Button(String xpath, String param, int index)- конструктор инициализирует кнопку с заданным xPath, значением атрибута и индексом элемента.

Методы:

- public void click() - метод нажатия на кнопку, перед нажатием проверяет, что элемент видим
- public void clickWithmouseOver(SelenideElement elementToMouseOver) - метод нажатия на кнопку, которая появляется только при наведении крусора на другой элемент
- public void hover() - метод наведения курсора на кнопку
- public boolean exists() - метод проверки существования кнопки на странице

Также содержит методы для создания объектов Button, по различным элементам и их значениям:

- public static Button byId(String id)
- public static Button byClass(String className)
- public static Button byClass(String className, int index)
- public static Button byData(String data)
- public static Button byDivNotHtmlButton(String className)
- public static Button byDataTestHtmlA(String dataTest)
- public static Button byDataTestHtmlDiv(String dataTest)
- public static Button byFullXPath(String xPath)
- public static Button byAriaPressedAndDataTestContains(String containsWord)

Класс DivButton - класс расширяет функциональность класса Button и предназначен для работы с кнопками, реализованными с использованием HTML - элемнта , наследуется от Button.

Поля:

- private static final String CLASS_AND_TEXT_CONTAINS_XPATH - шаблон для формирования xPath, которы позволяет искать элемент по его классу и содержащемуся в нем тексту

Конструкторы:

- private DivButton(String xPath, String param) - конструктор, который инициализирует объект DivButton с заданным xPath и значением атрибута. Устанавливает имя HTML-элемента как "div".
- private DivButton(String xPath) - конструктор, который инициализирует объект DivButton с заданным xPath.

Методы для создания divButton по различным параметрами:

- public static DivButton byDataTest(String dataTest)
- public static DivButton byClass(String className)
- public static DivButton byClassAndTextContains(String className, String text)

Класс LinkButton - предназначен для представления и взаимодействия с кнопками, которые реализованы как HTML-элементы <a>, также известные как

ссылки. Этот класс расширяет функциональность класса Button и добавляет специфические методы для создания объектов LinkButton на основе различных атрибутов.

Конструкторы:

- private LinkButton(String xPath, String param)

Методы:

- public static LinkButton byDataTest(String dataTest)
- public static LinkButton byClass(String className)

Класс Input - представление поля ввода текста на веб-странице и предоставляет методы для взаимодействия с ним. Наследует функциональность из класса BaseElement.

Конструкторы:

- private Input(String xPath) - создает Input по полному xPath.
- private Input(String xPath, String param) - создает Input по одному параметру.
- private Input(String xPath, String param, int index) - создает Input по одному параметру и индексу элемента. Используется когда на странице несколько элементов с одинаковым относительным xPath.
- private Input(String xPath, String param, String htmlName, boolean noHtml) - создает Input по одному параметру и названию веб-элемента, в случае, если он не является “input”.

Методы:

- public void fill(String value) - принимает строку. Заполняет поле этой строкой.

Методы для создания Input по различным параметрами:

- public static Input byId(String id)
- public static Input byName(String name)
- public static Input byClass(String className)
- public static Input byClassHtmlDiv(String className)

- public static Input byData(String data)
- public static Input byDataAndIndex(String data, int index)
- public static Input byPlaceholder(String placeholder)
- public static Input byFullXPath(String xPath)

Класс OtherElement - представляет все не описанные отдельно элементы веб-приложения. Наследует функциональность из класса BaseElement.

Конструкторы:

- public OtherElement(String fullXpath) - создает OtherElement по его xPath.

Методы:

- public void click() - метод нажатия на элемет, перед нажатием проверяет, что элемент видим.
- public void setValue(String value) - принимает строку, заполняет элемент ею.
- public boolean exists() - проверяет существует ли элемент на странице.
- public boolean has(WebElementCondition condition) - проверяет выполняется переданное условие для элемента.

Класс TextElement - представляет элемент, используемый только для получения текста. Наследует функциональность из класса BaseElement

Конструкторы:

- private TextElement(String xPath, String param, String elementType) - создает TextElement по заготовки xPath, типу веб-элемента, которым он является и значению атрибута.
- private TextElement(String xPath) - создает textElement по его xPath.

Методы для создания TextElement по различным параметрами:

- public static TextElement byDivClass(String className)

- public static TextElement byDivDataTest(String dataTest)
- public static TextElement byFullXPath(String xPath)

1.1.2 Pages

Пакет содержит классы, описывающие страницы и окна:

BasePage - абстрактный базовый класс для всех страниц веб-приложения. Предоставляет основные методы для работы с страницами. Методы, реализованные в классе:

- public <T extends BasePage> T refresh() - обновляет текущую страницу и возвращает объект страницы. Метод вызывает Selenide.refresh() для обновления страницы и затем возвращает текущую страницу, используя метод page().
- public <T extends BasePage> T page() - возвращает текущую страницу как объект типа T, где T - подкласс BasePage. Метод использует Selenide.page() для получения объекта страницы на основе текущего класса.
- public static <T extends BasePage> T page(Class<T> pageClass) - статический метод, возвращающий объект страницы типа T, где T - подкласс BasePage, на основе переданного класса pageClass. Метод использует Selenide.page() для создания объекта страницы указанного класса.

SideBarPages - наследуется от BasePage и представляет логику работы со страницей, на которой есть sidebar. Содержит следующие поля:

- private final Button createSpaceButton - кнопка при нажатии на которую открывается окно создания пространства.
- public static final String optionsSidebarButtonDatetest - строка, используемая для получения xPath кнопки “опции пространства” на боковой панели, форматируется для каждого пространства отдельно.

- private static final HashMap<String, Button> sidebarSpaceOptionButtons - хэш таблица, хранящая в качестве ключа название пространства, в качестве значения - кнопку для открытия опций пространства с данным названием. Выбрана именно HashMap, так как может использовать много пространств в одном тесте.
- private final String openSpaceSpanString - строка, используемая для получения xPath элемента span для открытия пространства, форматируется для каждого пространства отдельно.
- private static final Logger logger - логгер.

Методы, реализованные в классе:

- public FirstSpaceCreateWindow clickCreateSpaceButton() - еажимает кнопку (по умолчанию скрыта) создания нового пространства, для этого вызывает clickWithMouseOver, в качестве аргумента передает панель, на которой появится кнопка. Возвращает экземпляр класса первого окна для создания пространства.
- public SpaceOptionsWindow clickSidebarSpaceOptionButton(String spaceName) - принимает имя пространства, для которого надо открыть окно. Открывает окно опций для взаимодействия с выбранным пространством, для этого нажимает кнопку "опции" у выбранного пространства, используя clickWithMouseOver, а также HashMap для хранения названий пространств, если их будет несколько. Возвращает окно "опции для взаимодествия с пространствами" для выбранного пространства.
- public Optional<Boolean> isSpaceExists(String name) - принимает название пространства. Проверяет существует ли пространство с заданным именем, для этого проверяет существует ли кнопка для открытия пространства с заданным именем. Возвращает Optional, хранящее true, если пространство существует; false, если пространство не существует; null, если произошла ошибка.

- public Pair<String, SpacePage> createSpace(Space space) - принимает класс-обертку над параметрами пространства. Создает пространства, последовательно проделывая все требуемые шаги. В зависимости от аргумента может также задавать пространству описание, приватность, цвет, иконку. Возвращает Pair<описание ошибки при создании пространства (null, если ошибки не было) для дальнейшей обработки, экземпляр SpacePage (null, если произошла ошибка)>.
- public void deleteSpace(Space space) - принимает обертку над аргументами, класс Space, используется для получения имени пространства. Удаляет пространство с заданным именем, последовательно проделывая все требуемые шаги.
- public void openSpace(String name) - принимает имя пространства, открывает пространство по его имени.

SpacePage - наследуется от SpaceBarPage, представляет логику для работы со страницей рабочего пространства. Содержит следующие поля:

- private final Button openSpaceOptionsButton - кнопка для открытия опций пространства.
- public static final Logger logger - логгер.

Методы, реализованные в классе:

- public SpaceOptionsWindow clickOpenSpaceOptionsButton() - нажатие кнопки, открывающей окно с опциями взаимодействия с пространством. Возвращает экзмпляр класса SpaceOptionsWindow, описывающего логику работы с окном.
- public AllSpaceSettingsWindow openSpaceSettings() - открытие окна "все настройки" для пространства для этого последовательно открываются несколько окон: SpaceOptionsWindow, SpaceSettingsWindow и затем AllSpaceSettingsWindow. Возвращает экземпляр класса AllSpaceSettingsWindow, описывающего логику работы с окном.

- public String getSpaceDescription() - получения описания пространства из окна "все настройки", для этого вызывается метод класса SpacePage для получения окна, затем метод класса окна для получения описания. Возвращает описание пространства.
- public String getSpacePrivacySettings() - получения описания приватности пространства, используется openSpaceSettings(). Возвращает информация по приватности пространства для дальнейшей обработки.
- public Pair<String, String> getColorAndIcon() - В окне "все настройки" открывает окно выбора цвета и иконки, получает инфомарцию о выбранных. Возвращает Pair<Цвет, Иконка>.
- public List<String> getViews() - получает список всех view, которые есть в пространстве. Возвращает List<String> - все view в пространстве.

HomePage - наследуется от SideBarPage, представляет логику для работы с домашней страницей приложения. Содержит следующие поля:

- private final Button profileButton - кнопка профиля пользователя.
- private final Button taskButton - кнопка открытия окна с заданиями
- private final Button docsPageButton - кнопка для перехода на страницу документов.

Методы, реализованные в классе:

- public boolean checkButton() - проверяет, отображается ли кнопка профиля пользователя. Используется для проверки успешной авторизации.
- public TasksWindow clickTaskButton() - нажимает на кнопку открытия окна с заданиями. Возвращает экземпляр класса всплывающего окна с заданиями.
- public DocsPage clickDocsPageButton() - нажатие на кнопку перехода на страницу документов. Возвращает экземпляр класса DocsPage.

DocsPage - наследуется от SideBarPage, представляет логику для работы со страницей документов. Содержит следующие поля:

- private static final Logger logger - логгер.
- private final Button createDocButton - кнопка создания нового документа.
- private final Button selectAllButton - кнопка выбора всех документов.
- private final Button deleteButton - кнопка удаления выбранных документов.
- private final Button deselectButton - кнопка сброса выбора документов.
- private final Button archiveTabButton - кнопка перехода во вкладку таблицы с архивированными документами.

Методы, реализованные в классе:

- public DocCreationWindow clickCreateDocButton() - нажатие на кнопку создания нового документа. Возвращает экземпляр класса всплывающего окна создания документа.
- public void selectDocByIndex(int index) - нажатие на маркер выбора документа с индексом index (отсчет ведется с 1, начиная с верха таблицы).
- public void selectAllDocs() - нажатие на маркер выбора всех документов.
- public void clickDeselectButton() - нажатие на кнопку сброса выбора документов.
- public void clickDeleteButton() - нажатие на кнопку удаления выбранных документов.
- public int selectedDocsNumber() - возвращает количество выбранных документов.
- public DocEditingWindow clickEditDoc(int index) - нажатие на документ с индексом index (отсчет ведется с 1, начиная с верха

таблицы). Возвращает экземпляр класса всплывающего окна редактирования документа.

- public void switchToArchiveTab() - нажатие на вкладку архива.

BaseWindow - абстрактный базовый класс, от которого наследуются все окна.

Пакет windowsForSpaces содержит:

FirstSpaceCreateWindow - класс, реализующий работу с первым окном, появляющимся при создании пространства. В нем настраиваются название, описание, приватность, цвет и иконка (аватар). Класс содержит в себе следующие поля:

- private final Input spaceNameInput - поле для введения названия пространства.
- private final Input spaceDescriptionInput - поле для введения описания пространства.
- private final Button continueCreationButton - кнопка для прололжения создания пространства. Открывает следующее окно.
- private final TextElement errorContainer - элемент, хранящий описание ошибки в названии пространства.
- private final Button switchAccessibilityButton - кнопка для изменения настроек приватности.
- private final Button openAndCloseCustomiseButton - кнопка для открытия кастомизации пространства.
- private final String pickColorString - строка, используемая для получения кнопка выбора цвета. Форматируется в зависимости от цвета.
- private final String pickIconString - строка, используемая для получения кнопка выбора иконки. Форматируется в зависимости от иконки.

- public static final Logger logger - логгер.

В классе реализованы следующие методы:

- public FirstSpaceCreateWindow() - Конструктор, логирует открытие окна.
- public void fillSpaceNameInput(String name) - принимает название пространства. Вводит название пространства в поле для названия.
- public void fillSpaceDescriptionInput(String description) - принимает описание пространства. Вводит описание пространства в поле для описания
- public SecondSpaceCreateWindow clickContinueCreationButton() - нажимает кнопку "continue", чтобы продолжить создание пространства; переходит к другому окну. Возвращает экземпляр класса второго окна для создания пространства.
- public boolean isError() - проверяет появилась ли информация об ошибке ввода названия. Возвращает true - ошибка появилась, false - ошибки нет.
- public String getCreateErrorText() - получает строку с информацией об ошибке, если ошибка произошла. Возвращает строку с информацией об ошибке, если ошибка произошла, иначе пустая строка.
- public void clickPrivacySettingsButton() - нажимает переключатель настройки приватности.
- public void clickOpenAndCloseCustomiseButton() - открвыает панель кастомизации названия и иконки пространства.
- public void closeCustomiseWindow() - закрывает окно кастомизации. Для этого нажимает "загрузить свою иконку", а затем закрывает появившееся окно. Используется для закрытия панели кастомизации, потому что лучшего решения найдено не было. Необходимо, так как иначе закрыть окно кастомизации методами Selenide не получается

- public void pickColor(String color) - принимает строку в формате “r, g, b”. Выбирает цвет по заданному набору rgb.
- public void pickIcon(String icon) - принимает название иконки. Выбирает иконку по её названию.

SecondSpaceCreateWindow - второе окно, появляющееся при создании пространства. В нём выбираются базовые настройки исходя из workflow пользователя. В классе реализованы следующие поля:

- private final Button createSpaceButton - кнопка для завершения создания пространства.
- private final String workFlowString - строка для получения кнопки выбора workflow, форматируется для каждого workflow отдельно.
- private final Logger logger - логгер.

В классе реализованы следующие методы:

- public SecondSpaceCreateWindow() - конструктор, логирует открытие окна.
- public void clickCreateSpaceButton() - нажимает кнопку завершения создания пространства.
- public void pickWorkFlow(int workflowNum) - принимает номер workflow. Выбирает Workflow из 4 предложенных вариантов.

SpaceOptionsWindow - окно, открывающееся при нажатии на опции пространства (три точки около его названия). Используется для открытия настроек и удаления пространства. Содержит поля:

- private final LinkButton deleteSpaceButton - кнопка для удаления пространства.
- private final LinkButton openSpaceSettingsButton - кнопка для открытия настроек пространства.
- public static final Logger logger - логгер.

В классе реализованы следующие методы:

- public SpaceOptionsWindow() - конструктор, логирует открытие окна.
- public DeleteSpaceWindow clickDeleteSpaceButton() - Нажимает кнопку "удалить пространство". Возвращает экземляр класса окна удаления пространства.
- public SpaceSettingsWindow clickOpenSpaceSettings() - Открывает настройки пространства. Возвращает экземпляр окна настроек пространства.

AllSpaceSettingsWindow - реализует взаимодействие с окном настроек пространства. Здесь можно изменить имя, описание, приватность цвет и иконку (аватар), а также посмотреть текущие их настройки. Содержит поля:

- private final TextElement descriptionContainer - веб-элемент, содержащий описание пространства.
- private final DivButton openShareSettingsButton - кнопка для открытия настроек приватности.
- private final DivButton openAvatarSettingsButton - кнопка для открытия настроек цвета и иконки.
- public static final Logger logger - логгер.

В классе реализованы следующие методы:

- public AllSpaceSettingsWindow() - конструктор, логирует открытие окна.
- public String getDescription() - получает описание пространства. Возвращает описание пространства.
- public ShareSettingsWindow openShareSettings() - открывает окно настроек приватности. Возвращает экземлпяр класса окна настроек приватности.
- public SpaceColorAndAvatarWindow openAvatarSettings() - открывает окно настроек цвета и иконки. Возвращает экземпляр окна настроек цвета и иконки.

DeleteSpaceWindow - окно, появляющееся для подтверждения удаления пространства. Содержит поля:

- private final Button submitDeleteSpaceButton - кнопка для подтверждения удаления пространства.
- public static final Logger logger - логгер.

В классе реализованы следующие методы:

- public DeleteSpaceWindow() - конструктор, логирует открытие окна.
- public DeleteSpaceWindow fillSubmitDeleteSpaceInput(String spaceName) - принимает название удаляемого пространства. Заполняет поле для подтверждения удаления пространства (Сайт требует ввести название пространства для подтверждения удаления). Возвращает this для удобной обработки в дальнейшем
- public void clickSubmitDeleteSpaceButton()

ShareSettingsWindow - реализует взаимодействия с окно, используемым для настроек приватности уже созданного пространства. тестах используется для получения этих данных. Содержит поля:

- private final TextElement privateSpaceText - элемент, хранящий текст с информаций о приватности пространства. В данном случае это текст на кнопке переключения настроек приватности.
- public static final Logger logger - логгер.

В классе реализованы следующие методы:

- public ShareSettingsWindow() - конструктор, логирует открытие окна.
- public String getPrivacyDescription() - получает информацию о приватности пространства. Возвращает описание приватности.

SpaceColorAndAvatarWindow - реализует взаимодействие с окном, используемым для настройки цвета и иконки (аватара) уже созданного пространства. В тестах используется для получения этих данных. Содержит поля:

- Button pickedColorButton - кнопка выбранного цвета.
- Button pickedIconButton - кнопка выбранной иконки.
- public static final Logger logger - логгер.

В классе реализованы следующие методы:

- public SpaceColorAndAvatarWindow() - конструктор, логирует открытие окна.
- public String getColor() - получает цвет пространства из атрибута style веб элемента. Возвращает цвет в формате "r, g, b".
- public String getIcon() - получает название иконки из атрибута веб элемента. Возвращает название иконки.

SpaceSettingsWindow - реализует взаимодействие с окном быстрой настройки пространства, имеет только несколько вариантов настройки. С его помощью открывается окно со всеми настройками. Содержит поля:

- private final LinkButton openAllSettingsButton - кнопка для открытия всех настроек пространства.
- public static final Logger logger - логгер.

В классе реализованы следующие методы:

- public SpaceSettingsWindow() - конструктор, логирует открытие окна.
- public AllSpaceSettingsWindow clickOpenAllSettingsButton() - открывает окно со всеми настройками пространства. Возвращает экземпляр класса окна со всеми настройками пространства.

Пакет windowForTasks - пакет содержит классы представляющие окна используемые для работы с задачами.

Класс TaskWindow - класс представляет собой окно задач, содержит методы для взаимодействия с элементами в окне задач.

Поля:

- private static final Logger logger - логгер

- private final Button todayButton - кнопка “Today” использует для наведения для последующего создания задачи, представленная объектом класса Button.

- private final Button addTaskButton - кнопка “+Task” для открытия окна создания задачи, представленная объектом класса Button.

- private final Button createTask - кнопка “Create Task” для создания задачи, представленная объектом класса Button.

- private final Button descriptionButton - кнопка “Description” для добавления описания задачи, представленная объектом класса Button.

- private final Button closeButton - кнопка “Close” для закрытия окна задач, представленная объектом класса Button.

- private final Button switchToDoneTasksButton - кнопка “Switch to Done Tasks” для перехода к списку выполненных задач, представленная объектом класса Button.

- private final Input descriptionField - поле ввода для описания задачи, представленное объектом класса Input.

- private final OtherElement taskNameField - поле ввода для названия задачи, представленное объектом класса OtherElement.

Методы:

- public boolean isTaskPresent(String taskName, Boolean taskAdded) - проверяет наличие задачи с заданным именем на странице.

- public boolean isMarkedDone(String taskName, Boolean taskAdded) - проверяет, отмечена ли задача как выполненная на странице.

- public TaskDescriptionWindow openTask(String taskName) - осуществляет клик по задаче с заданным именем, закрывает окно задачи и возвращает объект TasksDescriptionWindow для работы с описанием задачи.

- public void clickSwitchTasks() - выполняет переключение на отображение выполненных задач.

- public void clickCreateTaskButton() - кликает по кнопке для создания новой задачи.
- public void fillTaskNameField(String text) - заполняет поле для ввода названия задачи указанным текстом.
- public void clickAddTaskButton() - кликает по кнопке для добавления новой задачи.
- public void hoverTodayButton() - выполняет наведение курсора на кнопку "Today".
- public void clickDescriptionButton() - кликает по кнопке для открытия поля описания задачи.
- public void fillDescriptionField(String text) - заполняет поле для ввода описания задачи указанным текстом.

Класс TaskDescriptionWindow - класс представляет собой окно деталий задачи в веб-приложении, содержит методы для взаимодействия с описаниями и проверки их наличия.

Поля:

- private final Button priotiryButton - Кнопка для работы с приоритетом задачи.
- private final Button closeTaskDescriptionButton - Кнопка для закрытия окна деталей задачи.
- private final Button taskSettingsButton - Кнопка для открытия настроек задачи.
- private final Button deleteTaskButton - Кнопка для удаления задачи.
- private final Button markDoneButton - Кнопка для отметки задачи как выполненной.

Методы:

- public boolean isDescriptionPresent(String descriptionText, Boolean descriptionAdded) - Проверяет наличие описания задачи на странице.

- public boolean isPrioritySet(String priority, Boolean priorityAdded) - Проверяет, установлен ли приоритет для задачи.
- public void clickDoneButton() - Выполняет клик по кнопке "Done", отмечающей задачу как выполненную.
- public void clickPriorityButton() - Выполняет клик по кнопке для установки приоритета задачи.
- public void setPriority(String priority) - Устанавливает заданное значение приоритета для задачи.
- public void closeWindow() - Закрывает окно с описанием задачи.
- public void openTaskSettings() - Открывает настройки задачи.
- public void clickDeleteButton() - Выполняет клик по кнопке для удаления задачи.

Пакет windowsForDocs содержит всплывающие окна, появляющиеся при работе с документами:

DocCreationWindow - окно создания документа. Содержит поля:

- private static final Logger logger - логгер.
- private final Input titleInput - поле ввода заголовка (названия) документа.
- private final Button createButton - кнопка подтверждения создания документа.

В классе реализованы следующие методы:

- public void fillTitle(String docName) - заполнение названия документа строкой docName.
- public DocEditingWindow clickCreateButton() - нажатие на кнопку подтверждения создания документа. Возвращает объект всплывающего окна редактирования документа.

DocEditingWindow - окно редактирования документа. Содержит поля:

- private static final Logger logger - логгер.

- private final DivButton title - поле заголовка.
- private final DivButton closeButton - кнопка закрытия окна.

В классе реализованы следующие методы:

- public String getTitle() - возвращает название созданного документа.
- public void clickCloseButton() - нажатие на кнопку закрытия окна.

DocOptionsWindow - окно дополнительных опций для документа. Содержит поля:

- private static final Logger logger - логгер.
- private final Button duplicateButton - кнопка дублирования документа.
- private final Button archiveButton - кнопка перемещения документа в архив.

В классе реализованы следующие методы:

- public void clickDuplicateButton() - нажать на кнопку дублирования документа.
- public void clickArchiveButton() - нажать на кнопку перемещения документа в архив.

1.1.3 Tests

Пакет содержит классы, описывающие тесты:

BaseTests - абстрактный базовый класс для представления тестов. Этот класс служит основой для других тестовых классов, предоставляя им базовые настройки и методы для упрощения процесса написания и выполнения тестов, содержит следующие поля :

- BASE_URL - Базовый URL для открытия сайта ClickUp.
- PROPERTIES_PATH - Путь к файлу конфигурации, содержащему необходимые настройки и учетные данные.

- USER_EMAIL_PROPERTY - Ключ для получения значения email пользователя из файла конфигурации.
- USER_PASSWORD_PROPERTY - Ключ для получения значения пароля пользователя из файла конфигурации.
- CONFIG_TIMEOUT_PROPERTY - Ключ для получения значения таймаута ожидания элементов из файла конфигурации.
- CONFIG_PAGE_TIMEOUT_PROPERTY - Ключ для получения значения таймаута загрузки страницы из файла конфигурации.
- properties - Объект класса `Properties`, используемый для загрузки и хранения настроек из файла конфигурации.
- logger - Объект для логирования информации о выполнении тестов.

Также класс содержит следующие методы:

- static блок - Статический блок инициализации, который загружает настройки из файла конфигурации при загрузке класса. Если при чтении файла происходит ошибка, она обрабатывается и выводится стек вызовов.
- public void setUp() - Имеет аннотацию `@BeforeEach`, метод для настройки окружения перед каждым тестом. Включает настройку браузера (Chrome), его размеров, параметров работы, а также открытие базового URL. Таймауты ожидания элементов и загрузки страниц устанавливаются из значений, полученных из файла конфигурации.
- public void tearDown() - Имеет аннотацию `@AfterEach`, метод для завершения тестов и очистки окружения после каждого теста. Включает паузу и закрытие браузера.
- public HomePage auth() - Метод для авторизации на сайте ClickUp. Открывает страницу логина, вводит учетные данные пользователя

(email и пароль) и осуществляет вход на сайт. После успешной авторизации возвращает объект домашней страницы (`HomePage`).

LoginTest - класс тестирует функцию авторизации пользователя и проверяет, что после успешного входа на главной странице отображается иконка профиля, наследуется от BaseTest.

Пакет taskTests содержит классы, тестирующие различные функции для работы с задачами:

- CreateTaskTest - класс предназначен для тестирования создания задач, наследуется от BaseTest.
- CreateTaskWithDescriptionTest - класс предназначен для создания задач с описанием, наследуется от BaseTest.
- TasksWithPriorityTests - класс предназначен для проверки функциональности установки приоритетов задач, наследуется от BaseTest.
- DeleteTaskTest - класс предназначен для проверки функции удаления задач, наследуется от BaseTest.
- DoneTaskTest - класс предназначен для проверки функции пометки задачи как выполненой, наследуется от BaseTest.

Пакет docsTests содержит классы, тестирующие различные функции для работы с документами:

- ArchiveDocTest - класс тестирования перемещения документа в архив. Наследуется от BaseTest.
- CreateDocTest - класс тестирования создания документов. Наследуется от BaseTest.
- DeleteAllDocsTest - класс тестирования удаления всех документов. Наследуется от BaseTest.

- DeleteDocTest - класс тестирования удаления последнего документа. Наследуется от BaseTest.
- DuplicateDocTest - класс тестирования дублирования документов. Наследуется от BaseTest.

Пакет spaceTests содержит классы, тестирующие создание и удаление рабочих пространств.

- CreatePrivateSpaceTest - класс тестирования создания пространств с разными настройками приватности. Наследуется от BaseTest.
- CreateSpaceTest - класс тестирования создания пространств. Наследуется от BaseTest.
- CreateSpaceWithCustomIconAndColorTest - класс тестирования создания пространств с выбором цвета и иконки. Наследуется от BaseTest.
- CreateSpaceWithDescriptionTest - класс тестирования создания пространств с описанием. Наследуется от BaseTest.
- CreateWithCustomWorkFlow - класс тестирования создания пространств с выбором workflow. Наследуется от BaseTest.
- DeleteSpaceTest - класс тестирования удаления пространств по названию. Наследуется от BaseTest.

1.1.4 Utils

Пакет содержит классы, являющиеся утилитарными.

Класс ElementTypes представляет собой перечисление (enum), которое используется для определения различных типов HTML - элементов, содержит поле:

- private final String name - Приватное поле, содержащее строковое представление типа HTML-элемента.

Класс имеет конструктор:

- ElementTypes(String name) - Инициализирует перечисления заданным строковым значением типа HTML-элемента.

Класс имеет метод:

- public String toString() - Метод позволяет получать строку, ассоциированную с конкретным типом элемента.

Класс Space представляет собой обертку над аргументами для создания пространства. Содержит поля:

- private String name - название пространства.
- private String description - описание пространства.
- private boolean isPrivate - приватность пространства.
- private String color - цвет пространства. Строка в формате “r, g, b”
- private String icon - иконка пространства.
- private int workflow - workflow пространства.

Методы класса - конструкторы, геттеры и сеттеры поле.

1.2. UML

UML диаграмма созданных в ходе работы классов представлена на рисунке 1.

Более крупную версию диаграммы см. в приложении Б.

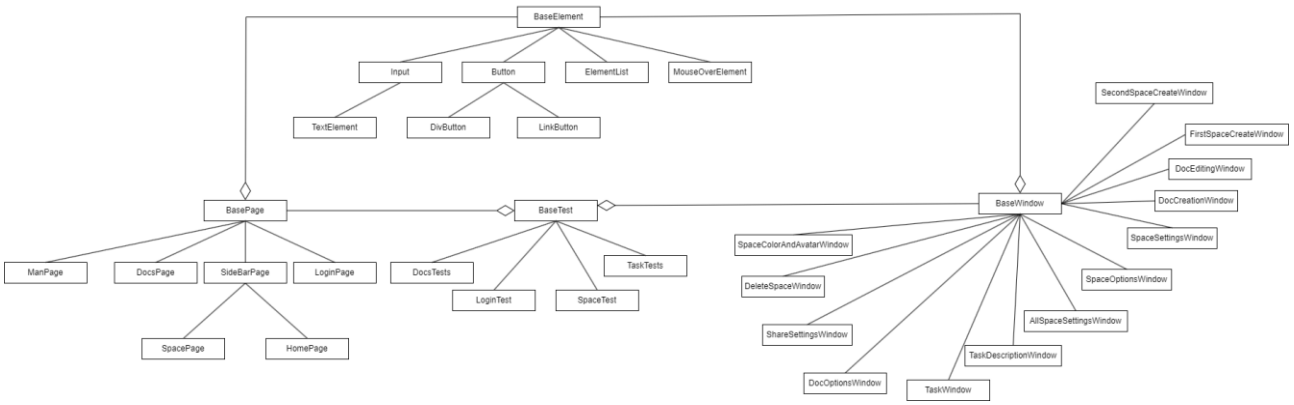

Рисунок 1 - UML диаграмма

2. ОПИСАНИЕ ТЕСТОВ

Чеклист тестов находится в приложении А

2.1. Тестирование создания и удаления задач, проверка различных функций

Рассмотрим тест выполняющий проверку корректности создания задачи:

0) Открытие сайта (см. рисунок 2)

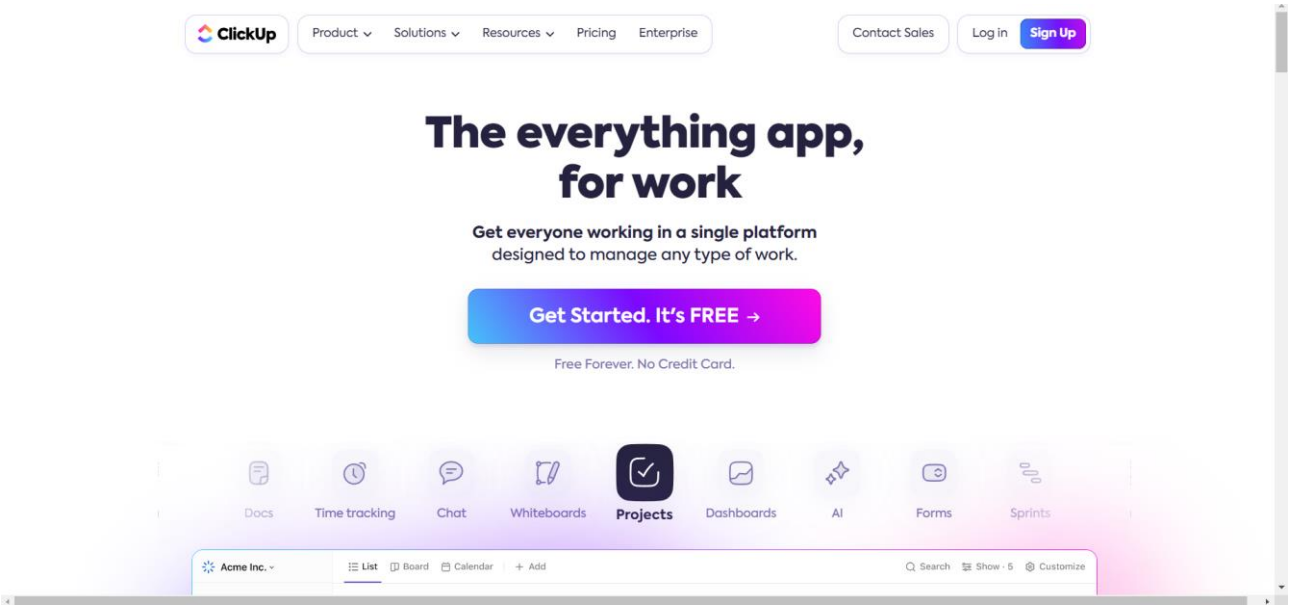

Рисунок 2 - Открытие сайта

1) Нажатие на кнопку ‘Login’ (см. рисунок 3)

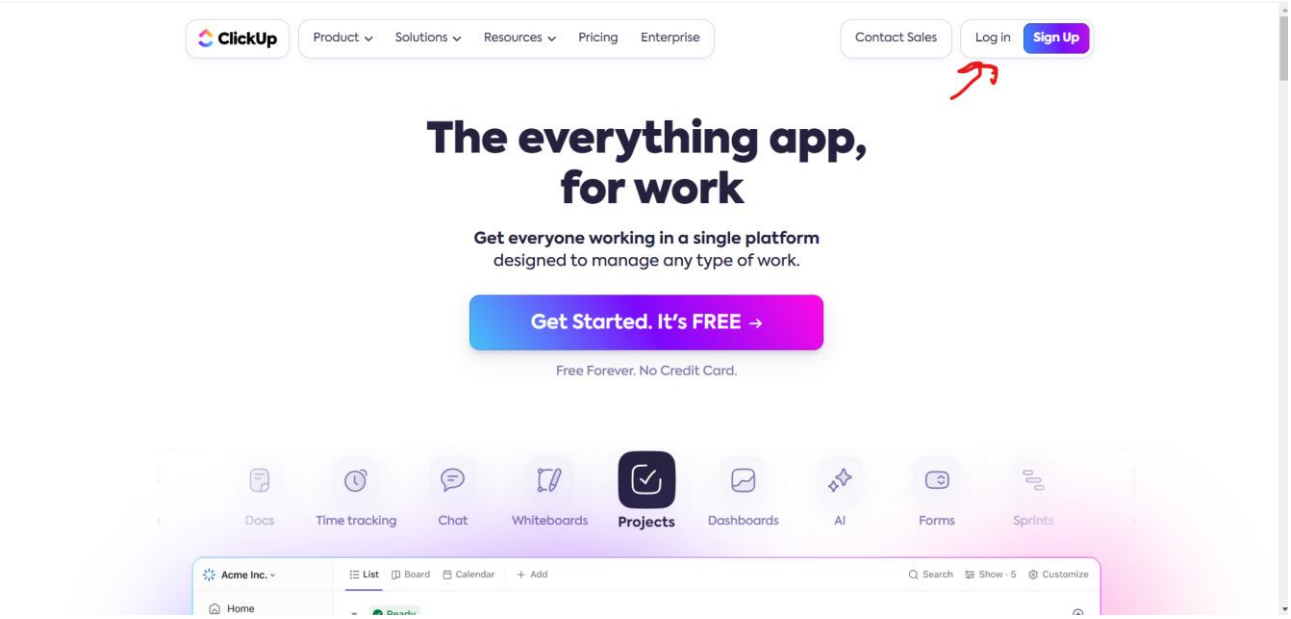

Рисунок 3 - Нажатие на кнопку ‘Login’

2) Авторизация пользователя на открывшейся странице (см. рисунок 4)

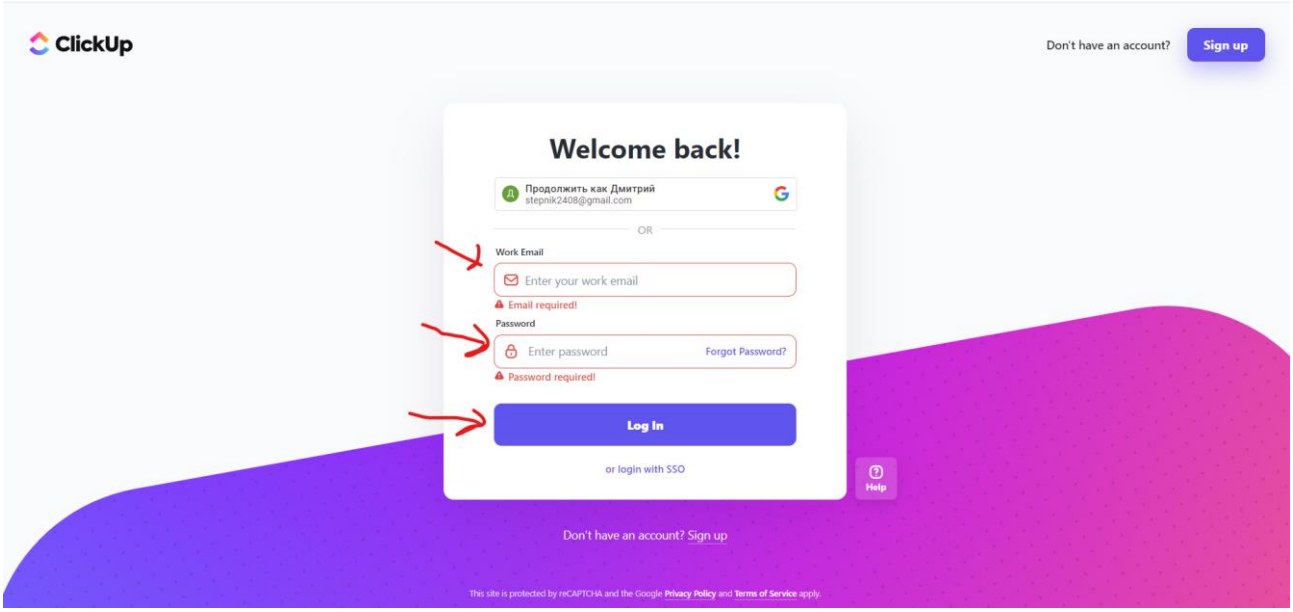

Рисунок 4 - Авторизация пользователя

3) На главной странице открытие окна с задачами путём нажатия на кнопку ‘Open My Tasks’ (см. рисунок 5)

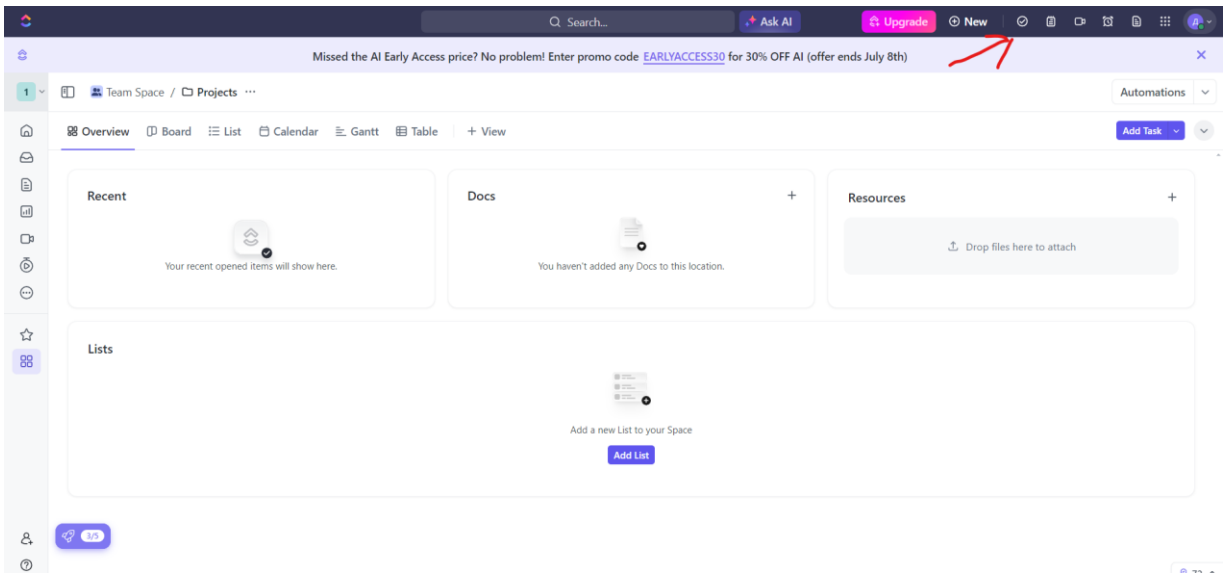

Рисунок 5 – Нажатие на кнопку ‘Open My Tasks’

4) В открывшемся окне наведение курсора на кнопку ‘Today’ (см. рисунок 6)

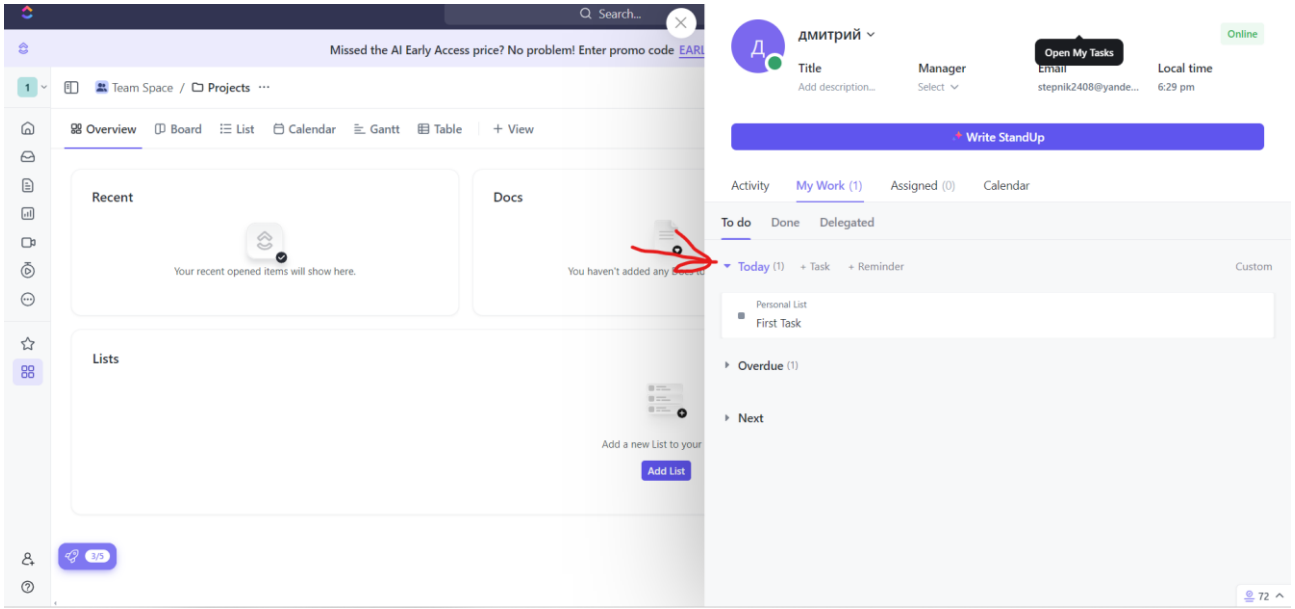

Рисунок 6 - Наведение курсора на кнопку ‘Today’

5) Нажатие на кнопку ‘+ Task’ (см. рисунок 7)

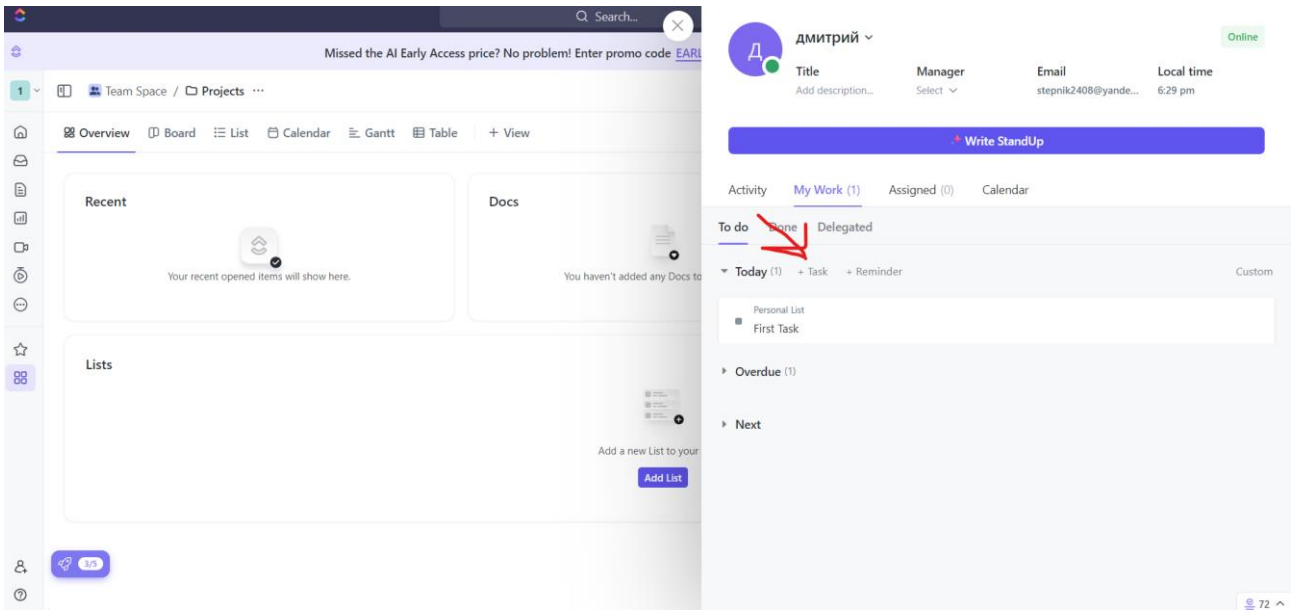

Рисунок 7 - Нажатие на кнопку ‘+ Task’

6) В открывшемся окне добавить заданное название задачи в поле ‘Task name’ (см. рисунок 8)

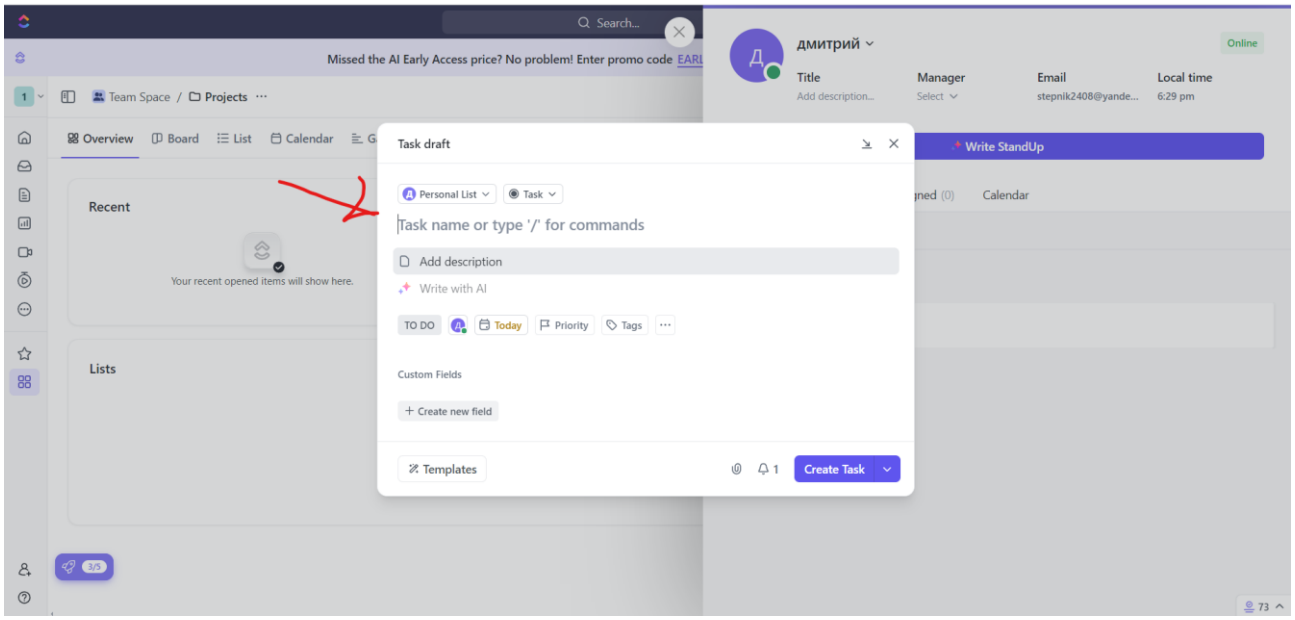

Рисунок 8 – Добавление названия задачи

7) Нажатие на кнопку ‘Create Task’ (см. рисунок 9)

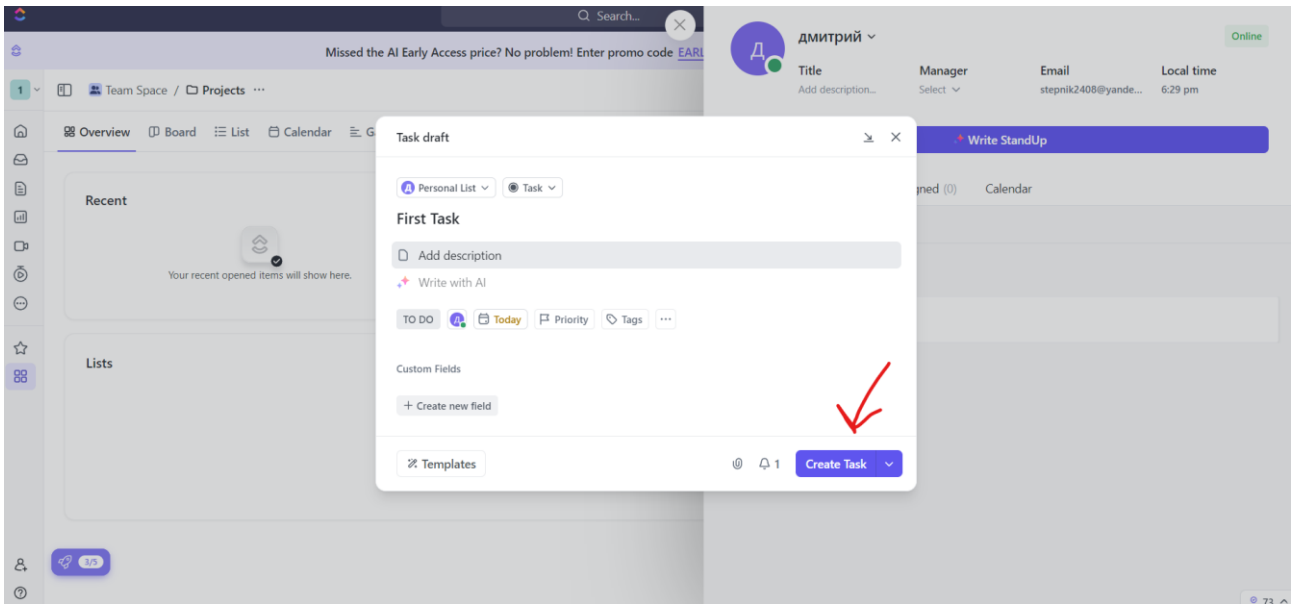

Рисунок 9 - Нажатие на кнопку ‘Create Task’

8) Проверить, что появился новый блок задачи с указанным названием (см. рисунок 10)

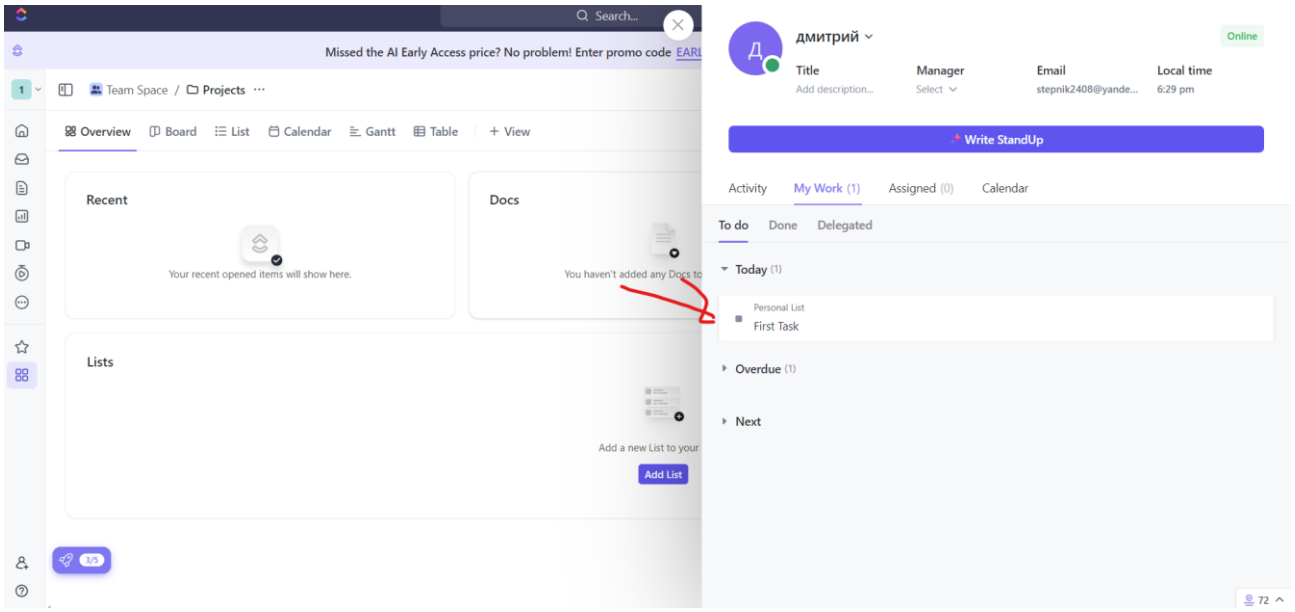

Рисунок 10 – Проверка появления нового блока задачи

В итоге получаем созданную задачу с указанным названием.

Полученные логи:

2024-07-05 18:18:05 [main] INFO tests.BaseTest - main page is opened

2024-07-05 18:18:08 [main] INFO tests.BaseTest - login page is opened

2024-07-05 18:18:09 [main] INFO pages.LoginPage - email field is filled: ****

2024-07-05 18:18:09 [main] INFO pages.LoginPage - password field is filled: ****

2024-07-05 18:18:09 [main] INFO pages.LoginPage - submit login button is clicked

Task window opened

2024-07-05 18:18:13 [main] INFO pages.popUpWindows.windowForTasks.TasksWindow - Hover to today button

2024-07-05 18:18:14 [main] INFO pages.popUpWindows.windowForTasks.TasksWindow - Task added

2024-07-05 18:18:14 [main] INFO pages.popUpWindows.windowForTasks.TasksWindow - Task filled with text - First Task

2024-07-05 18:18:14 [main] INFO pages.popUpWindows.windowForTasks.TasksWindow - Create task button clicked

tear down

2.2. Тестирование создания и удаления рабочих пространств с разными параметрами

Рассмотрим тест выполняющий проверку корректности создания рабочего пространства.

0) Открытие сайта (см. рисунок 11)

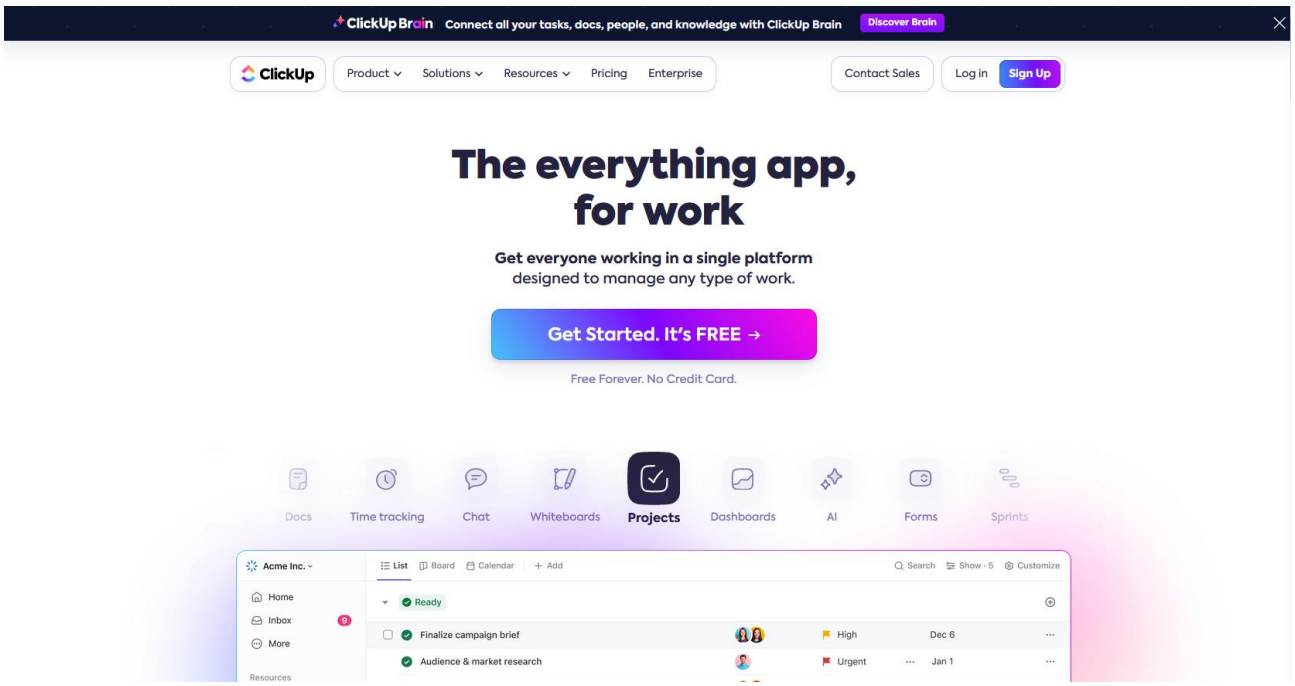

Рисунок 11 - Открытие сайта

1) Нажатие на кнопку ‘Login’ (см. рисунок 12)

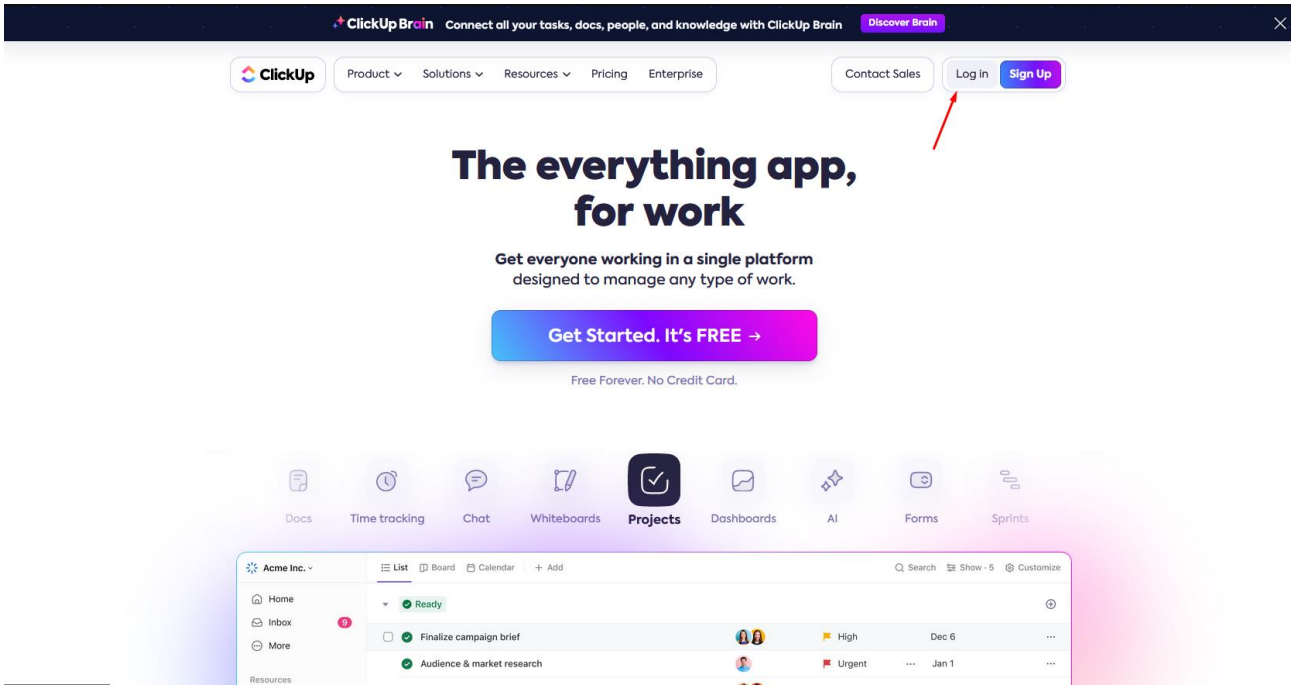

Рисунок 12 - Нажатие на кнопку ‘Login’

2) Авторизация пользователя на открывшейся странице (см. рисунок 13)

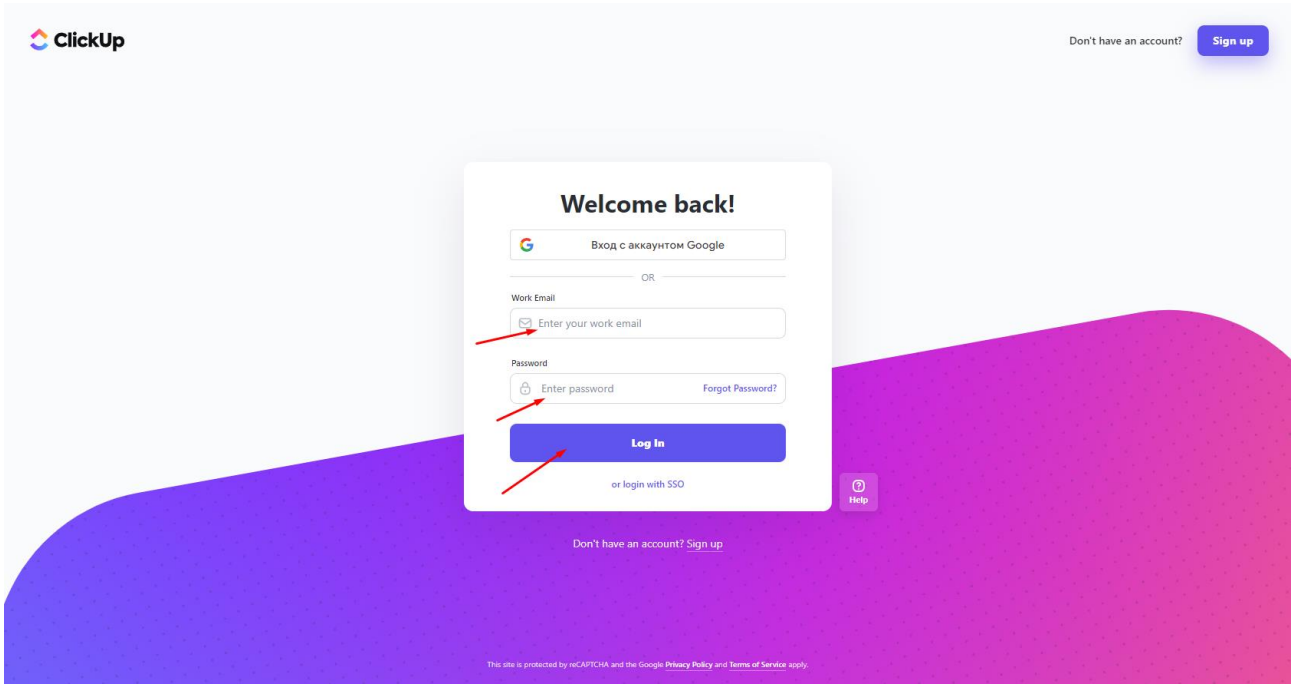

Рисунок 13 - Авторизация пользователя

3) Нажатие кнопки “+” (“New Space”) на боковой панели с пространствами (см. рисунок 14)

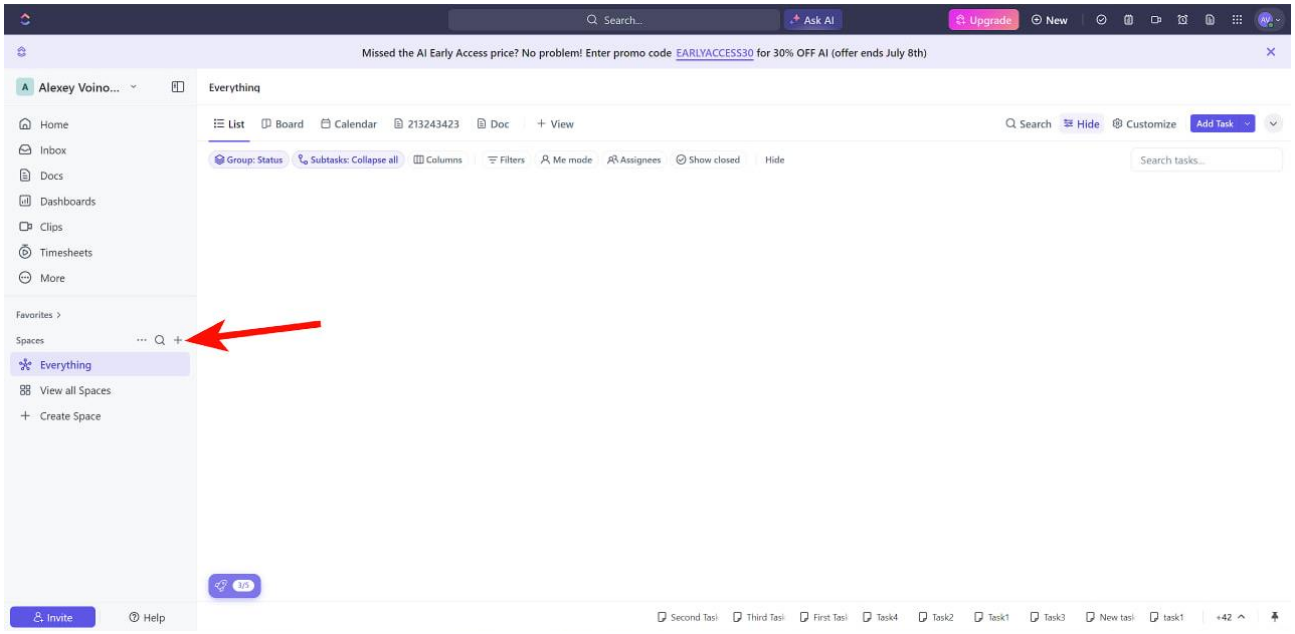

Рисунок 14 - Нажатие кнопки “+”

4) В открывшемся окне ввести название пространства (см. рисунок 15)

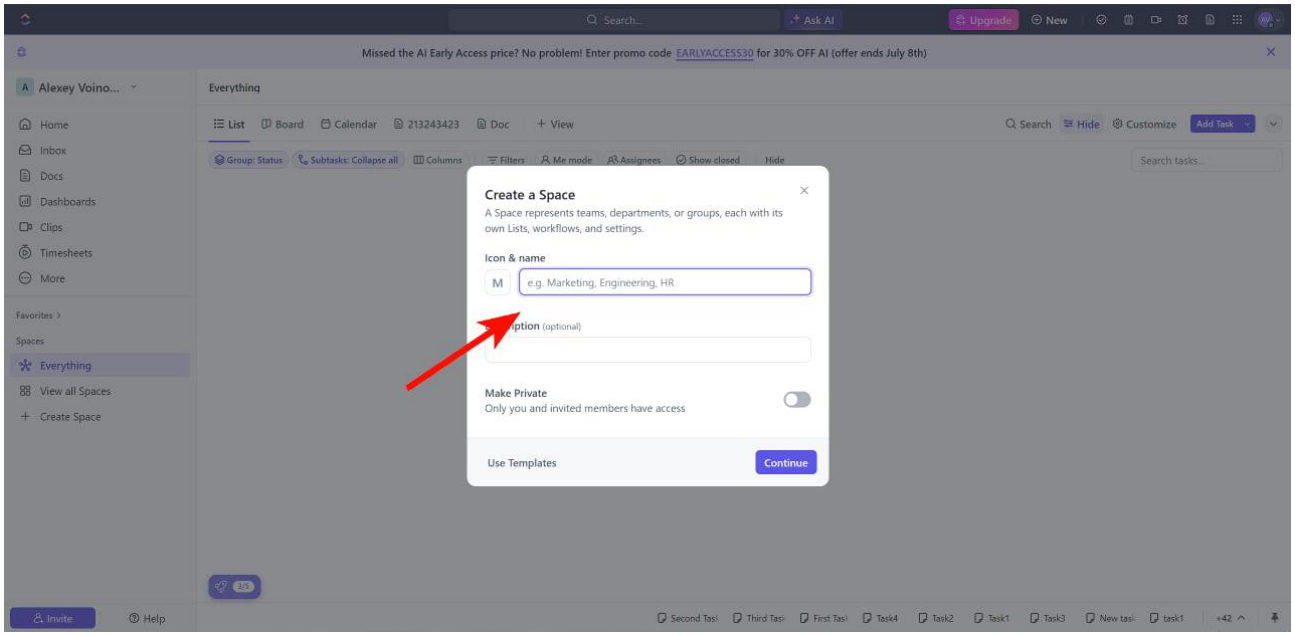

Рисунок 15 – Ввод названия пространства

5) Нажатие “Continue” (см. рисунок 16)

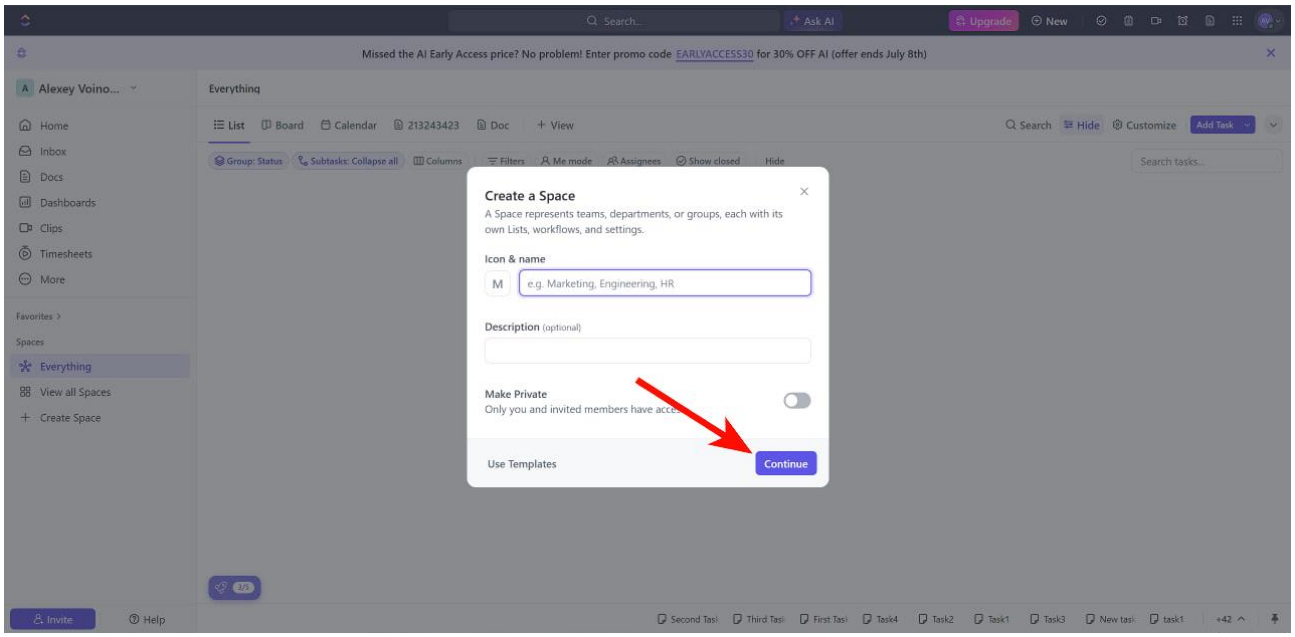

Рисунок 16 - Нажатие “Continue”

6) В открывшемся выбрать workflow “Starter” (см. рисунок 17)

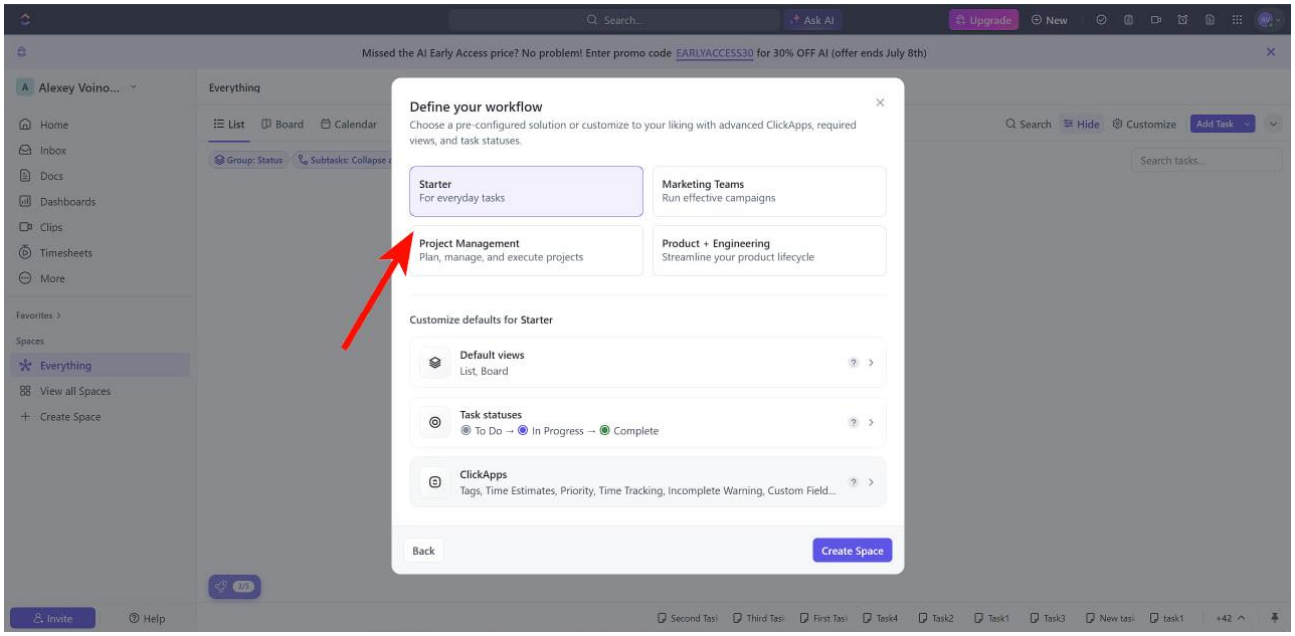

Рисунок 17 – Выбор workflow “Starter”

7) Нажатие “Create Space” (см. рисунок 18)

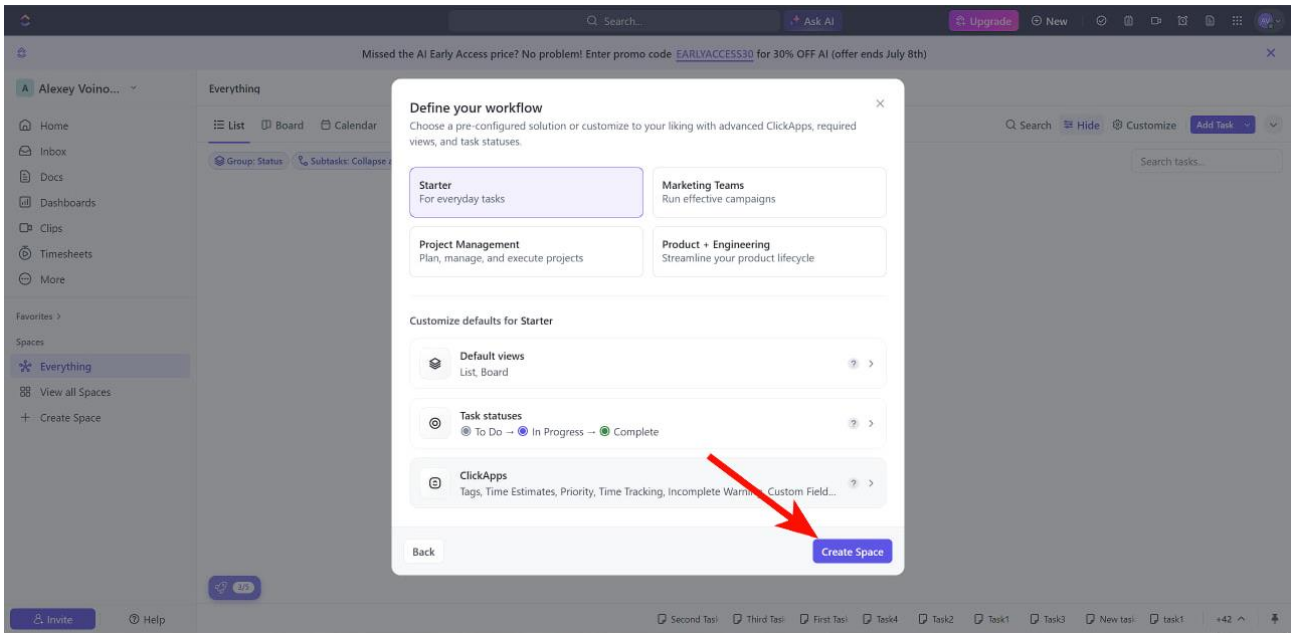

Рисунок 18 - Нажатие “Create Space”

8) Проверить, что пространство с заданным именем было создано и отображается на панели с пространствами (см. рисунок 19)

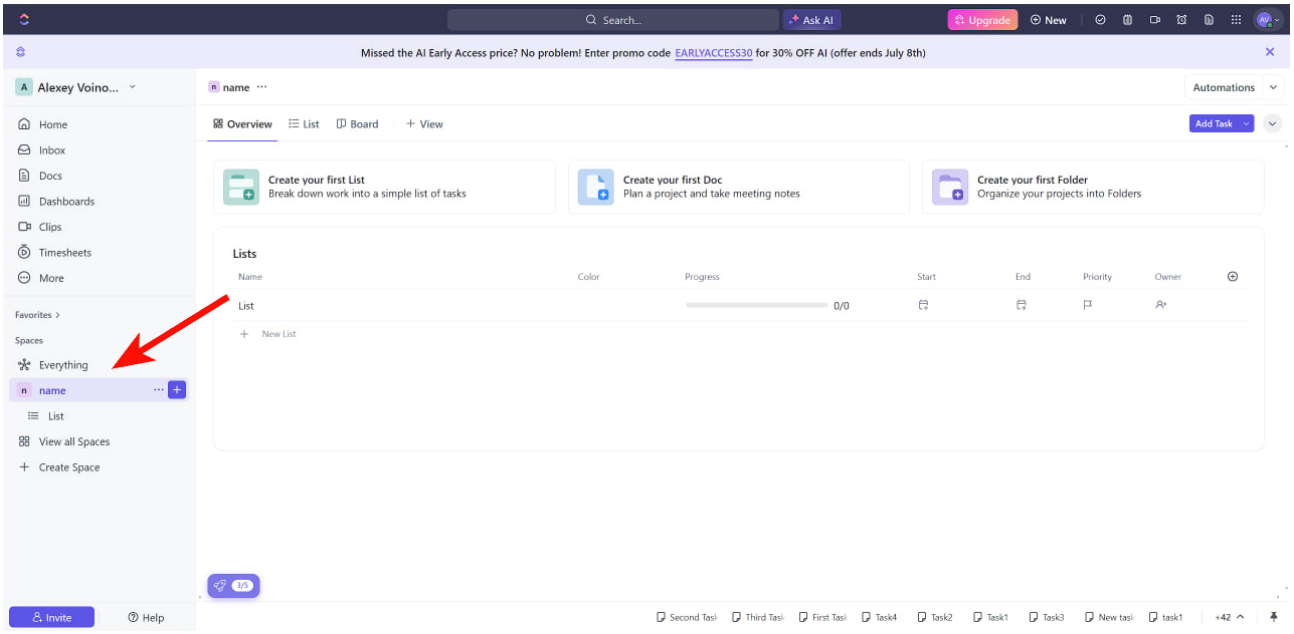

Рисунок 19 – Проверка создания пространства

В итоге получаем рабочее пространство с заданным именем.

Полученные логи:

2024-07-05 20:03:38 [main] INFO tests.spaceTests.CreateSpaceTest - create space test starts

2024-07-05 20:03:38 [main] INFO tests.BaseTest - main page is opened

2024-07-05 20:03:46 [main] INFO tests.BaseTest - login page is opened

2024-07-05 20:03:46 [main] INFO pages.LoginPage - email field is filled: ***

2024-07-05 20:03:47 [main] INFO pages.LoginPage - password field is filled: ***

2024-07-05 20:03:47 [main] INFO pages.LoginPage - submit login button is clicked

2024-07-05 20:03:57 [main] INFO pages.SideBarPage - create new space button clicked, creation window opened

2024-07-05 20:03:57 [main] INFO pages.popUpWindows.windowsForSpaces.createSpace.FirstSpaceCreateWindow - first creation window is opened

2024-07-05 20:03:57 [main] INFO pages.popUpWindows.windowsForSpaces.createSpace.FirstSpaceCreateWindow - name of space is entered, name = name

2024-07-05 20:03:57 [main] INFO pages.popUpWindows.windowsForSpaces.createSpace.FirstSpaceCreateWindow - description of space is entered, description = empty description

2024-07-05 20:03:57 [main] INFO pages.popUpWindows.windowsForSpaces.createSpace.FirstSpaceCreateWindow - continue button clicked

2024-07-05 20:03:57 [main] INFO pages.popUpWindows.windowsForSpaces.createSpace.SecondSpaceCreateWindow - second creation window is opened

2024-07-05 20:04:03 [main] WARN elements.BaseElement - element by xPath //div[@class='error-message ng-star-inserted']//span[@class='ng-star-inserted'] not found

2024-07-05 20:04:03 [main] INFO pages.popUpWindows.windowsForSpaces.createSpace.FirstSpaceCreateWindow - name is right

2024-07-05 20:04:03 [main] INFO pages.popUpWindows.windowsForSpaces.createSpace.SecondSpaceCreateWindow - worklow picked, number: 0

2024-07-05 20:04:03 [main] INFO pages.popUpWindows.windowsForSpaces.createSpace.SecondSpaceCreateWindow - create space button clicked

2024-07-05 20:04:08 [main] INFO pages.SideBarPage - second creation window is closed

2024-07-05 20:04:08 [main] INFO pages.SideBarPage - space name name

2024-07-05 20:04:08 [main] INFO pages.SideBarPage - space name is exists: true

tear down

В логах есть один warn. Он появился из-за попытки метода isError() класса FirstSpaceCreateWindow узнать отображается ли элемент, содержащий ошибку. Так как ошибки не было, элемент не отобразился. Следовательно метод isDisplayed() класса BaseElement бросил исключение, которое было обработано выедением warn и возвращение false, что значит - ошибки не произошло.

2.3. Тестирование создания, удаления и перемещения документов.

Рассмотрим тест выполняющий проверку корректности перемещения документа в архив:

0) Открытие сайта (см. рисунок 20)

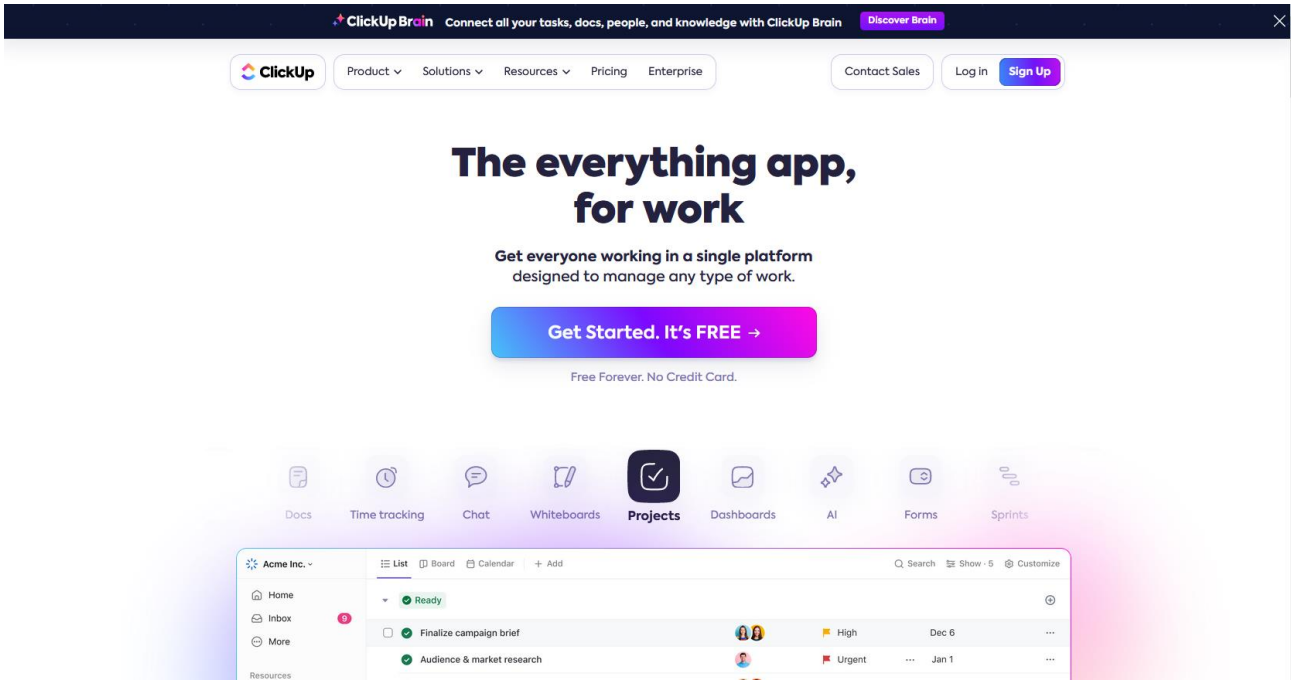

Рисунок 20 - Открытие сайта

1) Нажатие на кнопку ‘Login’ (см. рисунок 21)

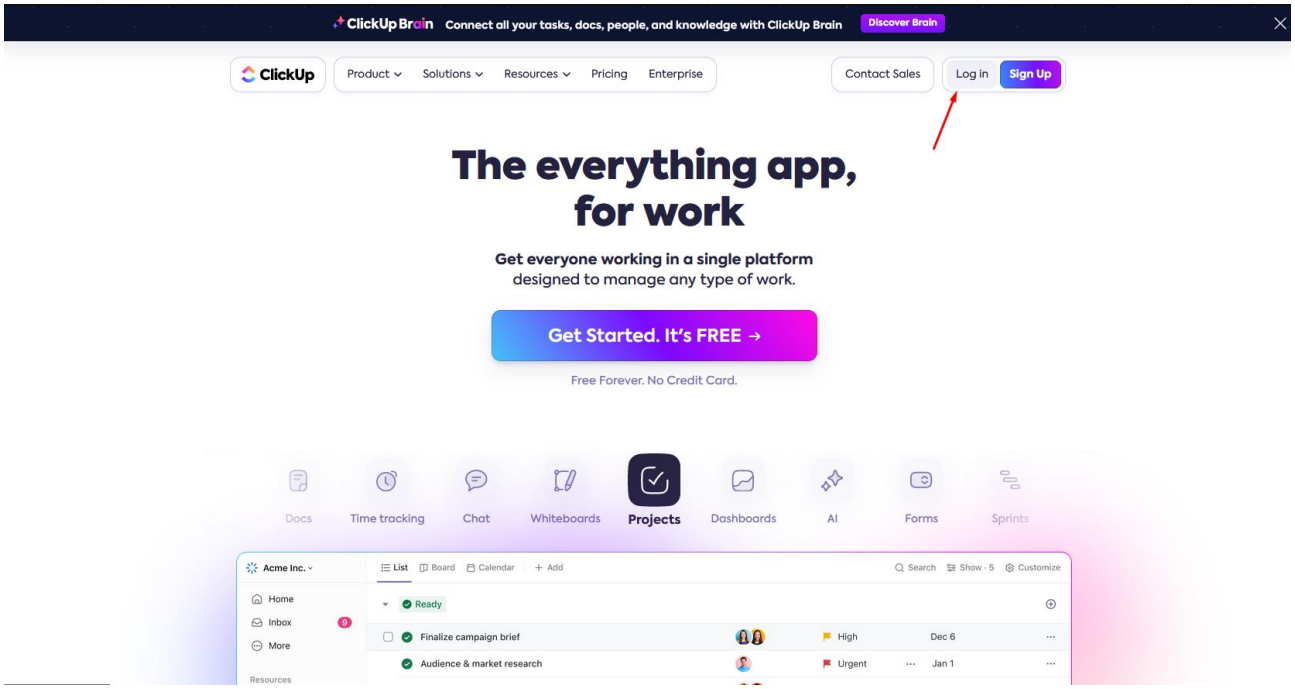

Рисунок 21 - Нажатие на кнопку ‘Login’

2) Авторизация пользователя на открывшейся странице (см. рисунок 22)

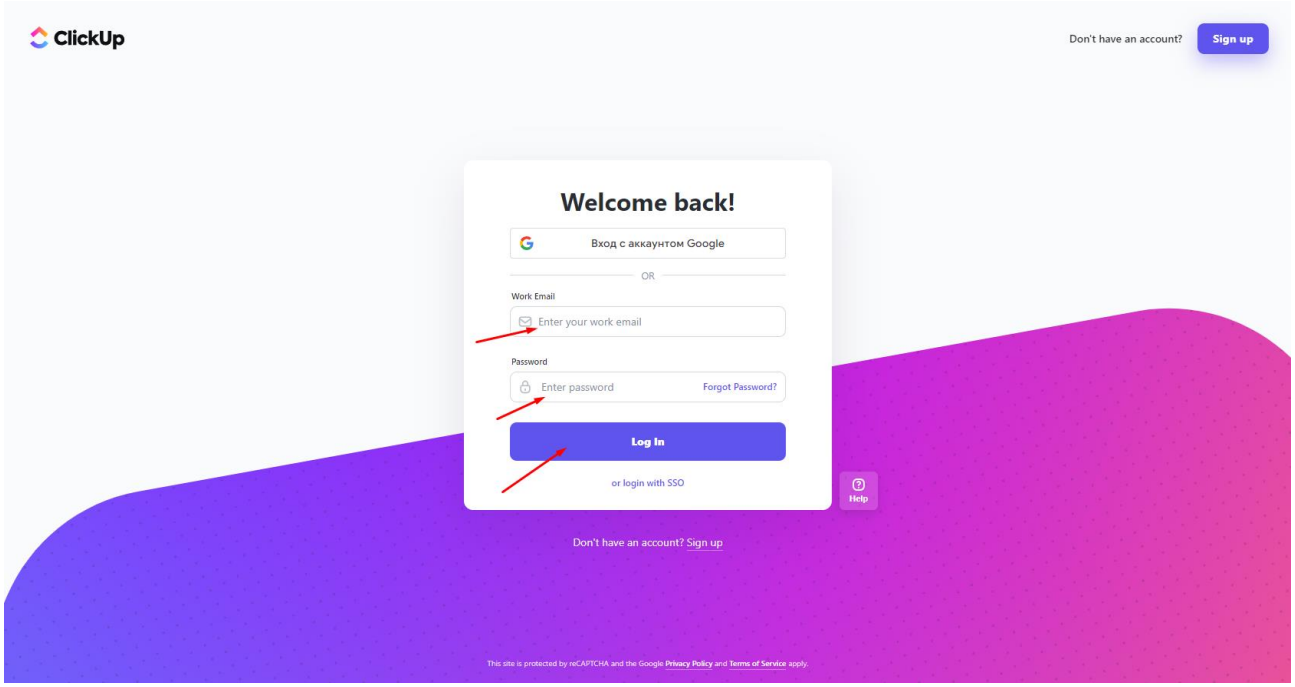

Рисунок 23 - Авторизация пользователя

3) Перейти на страницу ‘Docs’ (см. рисунок 23)

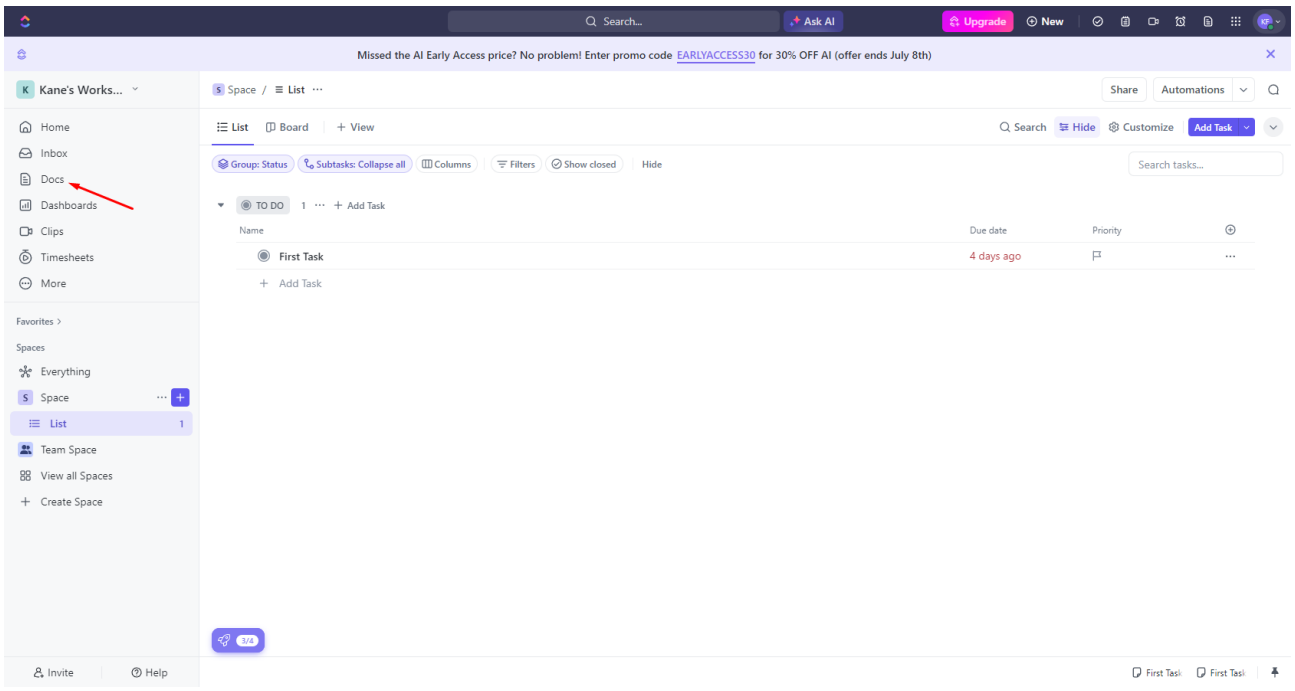

Рисунок 23 – Переход на страницу ‘Docs’

4) Для первого документа нажать на кнопку дополнительных опций (см. рисунок 24)

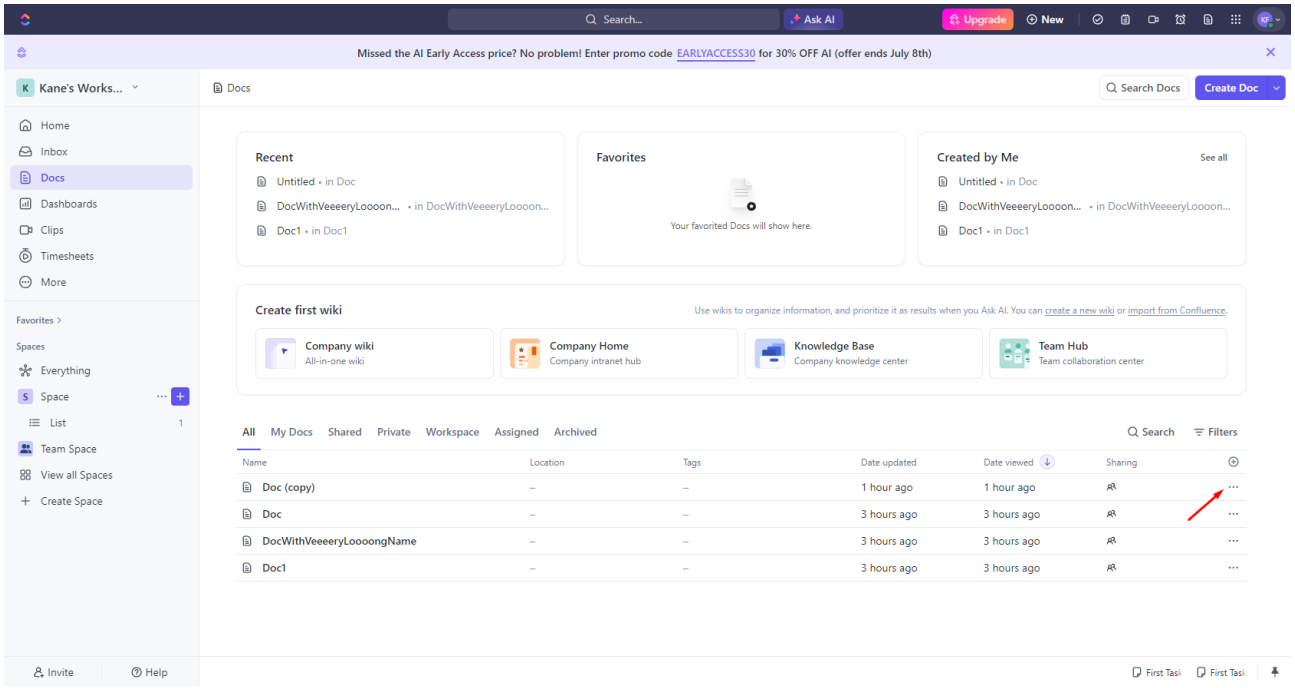

Рисунок 44 – Нажатие на кнопку опций для первого документа

5) Нажать на кнопку ‘Archive’ (см. рисунок 25)

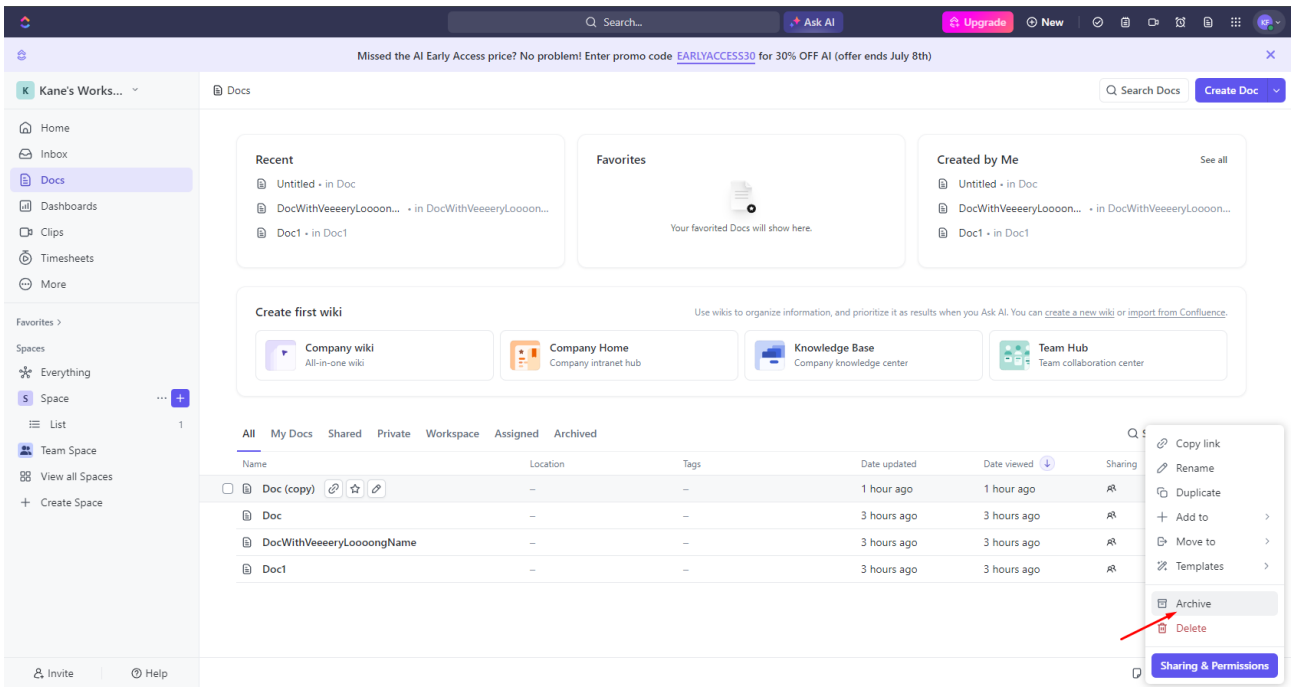

Рисунок 55 - Нажать на кнопку ‘Archive’

6) Переход во вкладку ‘Archived’, нажав на соответствующую кнопку (см. рисунок 26)

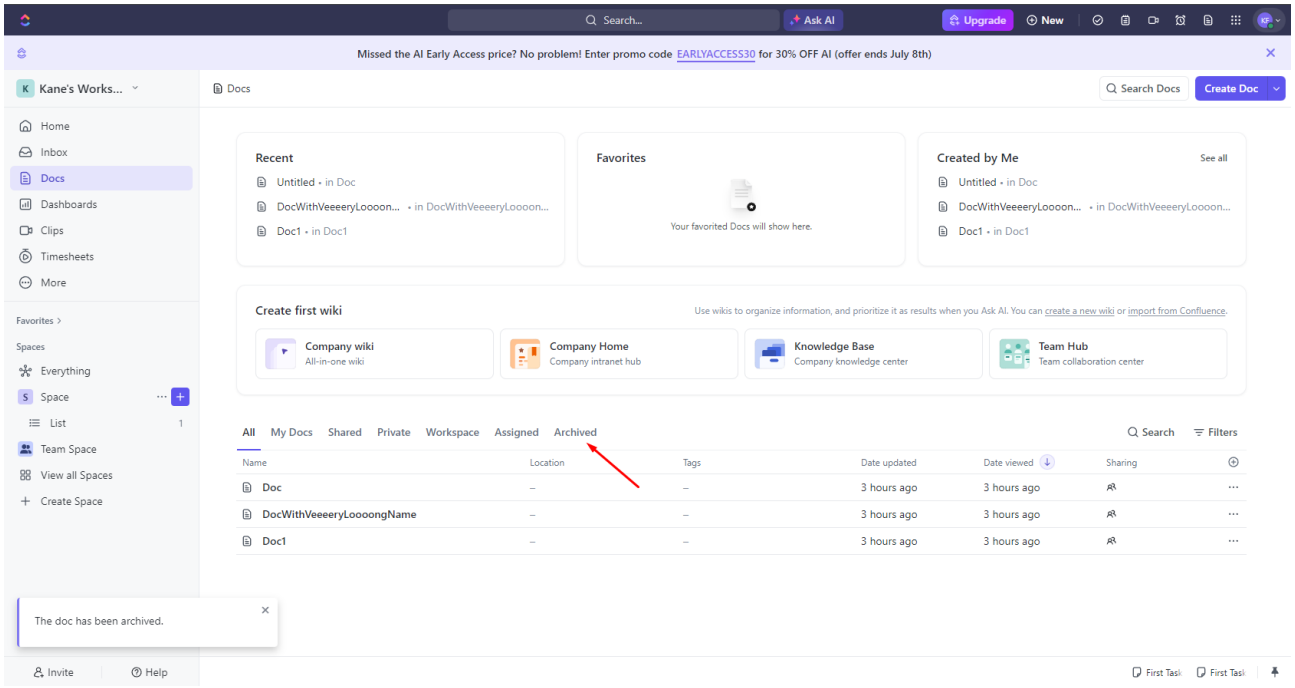

Рисунок 66 - Переход во вкладку ‘Archived’

7) Убедиться, что присутствует архивированный документ (см. рисунок 27)

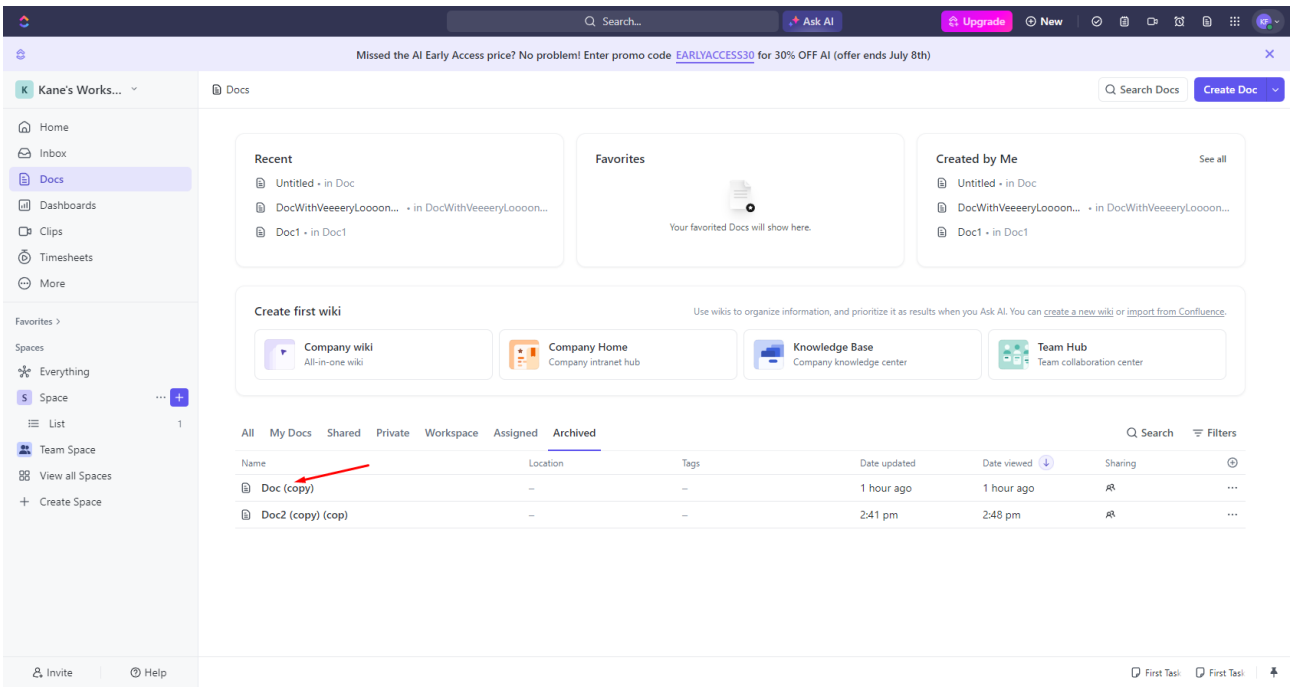

Рисунок 77 – Проверка присутствия архивированного документа

Полученные логи:

2024-07-05 20:51:54 [main] INFO tests.BaseTest - main page is opened

2024-07-05 20:52:06 [main] INFO tests.BaseTest - login page is opened

2024-07-05 20:52:08 [main] INFO pages.LoginPage - email field is filled: kaneflak@yandex.ru

2024-07-05 20:52:08 [main] INFO pages.LoginPage - password field is filled: kaneflak@yandex.ru

2024-07-05 20:52:08 [main] INFO pages.LoginPage - submit login button is clicked

2024-07-05 20:52:18 [main] INFO pages.DocsPage - Options button clicked for document with index 1

2024-07-05 20:52:18 [main] INFO pages.popUpWindows.windowsForDocs.DocOptionsWindow - Document archived

2024-07-05 20:52:18 [main] INFO pages.DocsPage - Switched to archive tab

ЗАКЛЮЧЕНИЕ

В ходе работы был изучен фреймворк для UI тестирования Selenide, фреймворк для Unit тестирования JUnit5, а также фреймворк для автоматизации сборки проекта maven. Также была изучена архитектура POM (Page Obejct Modal). Рассмотрены её составляющие: тесты, страницы, элементы, а также её главное ограничение - классы тестов ничего не знают о классах элементов страниц, что позволяет разрабатывать масштабируемые и поддерживаемые проекты.

На основе полученных знаний был написан проект по тестированию сайта Clickup. В ходе работы была создана основная архитектура проекта, основанная на Page Object Modal, представленная тремя пакетами: elements, pages, tests. Также были созданы классы пакета utils, используемые для упрощения разработки.

В целях создания легко поддерживаемой архитектуры, а также гарантии того, что тесты не будут взаимодействовать с ещё не открывающимися окнами был создан пакет pages.windows, реализующий логику работы с окнами.

В итоге разработаны тесты для трех функций сайта. Тесты помогают оценить работоспособность сайта, а также выявить его недостатки, как произошло в одном из тестов. В нём был найден недочет - отсутствие кнопки закрытия окна.

Задание практики было выполнено, а цель достигнута. Получены знания и опыт программирования на Java, а также разработки UI тестов с использованием Selenide, JUnit5 и maven.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Официальная документация Selenide // Selenide Documentation. URL: https://selenide.org/documentation.html (дата обращения: 28.06.2024).
2. Официальная документация JUnit5 // JUnit5 Documentation. URL: https://junit.org/junit5/docs/current/user-guide/ (дата обращения: 28.06.2024).
3. Код проекта на гитхабе // Github repository Element: https://github.com/ShevelevaAnna/SP_24_1

ПРИЛОЖЕНИЕ А

ЧЕКЛИСТ ТЕСТОВ

Тестируемый сайт - https://clickup.com

Таблица 1 – Чеклист тестов

№	Название	Входные данные	Описание действий	Результат
0	Общие тесты			
0.1	Логин	Email Password	1) Нажатие на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице путём заполнения полей указанными данными	Успешна выполнена авторизация пользователя с заданным паролем и почтой
1	Степанов Дмитрий (2303) - создание и удаление задач, проверка различных функций			
1.1	Создание задачи и проверка его отображения в окне задач	Данные для авторизации - Email Password Рандомное название задачи - Task name	1) Нажатие на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) На главной странице открытие окна с задачами путём нажатия на кнопку ‘Open My Tasks’ 4) В открывшемся окне наведение курсора на кнопку ‘Today’ 5) Нажатие на кнопку ‘+ Task’ 6) В открывшемся окне добавить заданное название задачи в поле ‘Task name’ 7) Нажатие на кнопку ‘Create Task’ 8) Проверить, что появился новый блок задачи с указанным названием	Успешно выполнено создание задачи.
1.2	Создание задачи с описанием и проверка	Данные для авторизации -	1) Нажатие на кнопку ‘Login’ 2) Авторизация пользователя на	Успешно выполнено создание задачи с описанием

№	Название	Входные данные	Описание действий	Результат
	отображения описания	Email Password Рандомное название задачи и описание задачи - Task name Task description	открывшейся странице 3) На главной странице открытие окна с задачами путём нажатия на кнопку ‘Open My Tasks’ 4) В открывшемся окне наведение курсора на кнопку ‘Today’ 5) Нажатие на кнопку ‘+ Task’ 6) В открывшемся окне добавить заданное название задачи в поле ‘Task name’ 7) В этом же окне нажатие на кнопку ‘Add description’ 8) В появившемся поле добавить заданное описание 9) Нажатие на кнопку ‘Create Task’ 10) Проверить, что при открытии задачи появилось поле с заданным описанием	
1.3	Создание задачи с заданным приоритетом и проверка установки приоритета	Данные для авторизации - Email Password Рандомное название задачи и доступный приоритет - Task name Priority	1) Нажатие на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) На главной странице открытие окна с задачами путём нажатия на кнопку ‘Open My Tasks’ 4) В открывшемся окне наведение курсора на кнопку ‘Today’ 5) Нажатие на кнопку ‘+ Task’ 6) В открывшемся окне добавить заданное название задачи в поле ‘Task name’ 7) Нажатие на кнопку ‘Create Task’ 8) Открытие только что созданной задачи путём нажатия на появившийся блок 9) В открывшемся окне нажатие на	Успешно выполнено создание задачи с заданным приоритетом

№	Название	Входные данные	Описание действий	Результат
			кнопку ‘Priority’ 10) Установка приоритета путём нажатия на кнопку соответствующую заданному приоритету 11) Проверить, что при открытии описания задачи появился заданный приоритет	
1.4	Удаление созданного задания и проверка корректности удаления	Данные для авторизации - Email Password Рандомное название задачи- Task name	1) Нажатие на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) На главной странице открытие окна с задачами путём нажатия на кнопку ‘Open My Tasks’ 4) В открывшемся окне наведение курсора на кнопку ‘Today’ 5) Нажатие на кнопку ‘+ Task’ 6) В открывшемся окне добавить заданное название задачи в поле ‘Task name’ 7) Нажатие на кнопку ‘Create Task’ 8) Открытие только что созданной задачи путём нажатия на появившийся блок 9) В открывшемся окне нажатие на кнопку ‘Task settings’ 10) Нажатие на кнопку ‘Delete’ в выпадающем окне 11) Проверить, что в окне с задачами, удалённой задачи нет	Успешно выполнено удаление задачи с указанным названием
1.5	Создание задачи и установка статуса “Complete” у созданной задачи	Данные для авторизации - Email Password	1) Нажатие на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) На главной странице открытие окна с задачами путём нажатия на	Успешно выполнена установка статуса “Complete” у задачи

№	Название	Входные данные	Описание действий	Результат
		Рандомное название задачи - Task name	кнопку ‘Open My Tasks’ 4) В открывшемся окне наведение на кнопку ‘Today’ 5) Нажатие на кнопку ‘+ Task’ 6) В открывшемся окне добавить заданное название задачи в поле ‘Task name’ 7) Нажатие на кнопку ‘Create Task’ 8) Открытие только что созданной задачи путём нажатия на появившийся блок 9) В открывшемся окне нажатие на кнопку ‘Mark complete’ 10) Закрытие окна путём нажатия на кнопку ‘Close window’ 11) Открытие окна с задачами и переход к выполненным задачам нажатием кнопки ‘Done’ 12) Проверить, что в списке выполненных задач появилась указанная задача	
2	Репкин Вадим (2303) - создание, удаление и редактирование документов			
2.1	Создание документа с заданным названием	Email Password Document name	1) Нажать на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) Переход на страницу документов путем нажатия на кнопку “Docs” 4) Нажать на кнопку “Create a Doc” для создания документа 5) Заполнить название документа в открывшемся окне 6) Нажать на кнопку “Create Doc” 7) Проверить, что создался документ с заданным именем	Успешно выполнено создание документа с заданным названием
2.2	Удаление последнего созданного документа	Email Password	1) Нажать на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) Переход на страницу	Успешно выполнено удаление последнего созданного документа

№	Название	Входные данные	Описание действий	Результат
			документов путем нажатия на кнопку “Docs” 4) Навестись на заголовок таблицы и выбрать все документы, нажав на появившийся флажок 5) Запомнить количество файлов, указанное во всплывшем меню 6) Нажать на кнопку “Deselecte All” для отмены выбора всех документов 7) Навестись на первую строку таблицы и выбрать документ, нажав на появившийся флажок 8) Нажать на кнопку “Delete” во всплывшем меню 9) Навестись на заголовок таблицы и выбрать все документы, нажав на появившийся флажок 10) Запомнить количество файлов, указанное во всплывшем меню 11) Проверить, что количество документов уменьшилось на один	
2.3	Удаление всех документов	Email Password	1) Нажать на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) Переход на страницу документов путем нажатия на кнопку “Docs” 4) Навестись на заголовок таблицы и выбрать все документы, нажав на появившийся флажок 5) Нажать на кнопку “Delete” во всплывшем меню 6) Проверить, что таблица пуста	Успешно выполнено удаление всех документов
2.4	Архивация документа	Email Password	1) Нажать на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) Переход на страницу документов путем нажатия на кнопку “Docs” 4) Нажать на кнопку дополнительного меню “...” в первой строке таблицы 5) Нажать на кнопку “Archive” в появившемся окне 6) Нажать на вкладку “Archive” в заголовке таблицы	Успешно выполнена архивация документа

№	Название	Входные данные	Описание действий	Результат
			7) Проверить, что в данной вкладке появился архивируемый документ	
2.5	Дублирование документа	Email Password	1) Нажать на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) Переход на страницу документов путем нажатия на кнопку “Docs” 4) Нажать на кнопку дополнительного меню “...” в первой строке таблицы 5) Нажать на кнопку “Duplicate” в появившемся окне 6) Проверить, что появился новый документ-копия (<название дублируемого документа> + “(copy)”)	Успешно выполнено дублирование документа
3	Войнов Алексей (2300) - создание и удаление рабочих пространств с разными параметрами			
3.1	Создание пространства с заданным именем и проверка его наличия	Email Password Space name	1) Нажатие на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) Нажатие кнопки “+” (“New Space”) на боковой панели с пространствами 4) В открывшемся окне ввести название пространства 5) Нажать “Continue” 6) В открывшемся выбрать workflow “Starter” 7) Нажать “Create Space” 8) Проверить, что пространство с заданным именем было создано и отображается на панели с пространствами	Успешно выполнено создание пространства с заданным именем
3.2	Создание пространства с заданным именем и описанием, проверка корректности создания	Email Password Space name Space description	1) Нажатие на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) Нажатие кнопки “+” (“New Space”) на боковой панели с пространствами 4) В открывшемся окне ввести название и описание пространства 5) Нажать “Continue” 6) В открывшемся выбрать	Успешно выполнено создание пространства с заданным именем и описанием

№	Название	Входные данные	Описание действий	Результат
			workflow “Starter” 7) Нажать “Create Space” 8) Нажать кнопку с названием пространства на панели слева 9) На открывшейся странице работы с пространством нажать “Options” (кнопка с 3 точками) 10) В появившемся окне нажать “Space settings” 11) В появившемся окне нажать “All Space settings” 12) Сравнить значение поля “Space description” с заданным	
3.3	Удаление пространства по его имени и описанием, проверка корректности удаления	Email Password Space name	1) Нажатие на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) Если пространства с заданным именем не существует, создать его, чтобы гарантировать наличие 4) На боковой панели с пространствами навести курсор на требуемое пространство для появления кнопки “Options” (кнопка с 3 точками) 5) Нажать “Options” требуемого пространства 6) В появившемся окне нажать “Delete” 7) В открывшемся окне ввести название пространства и нажать Delete для подтверждения удаления 8) Проверить, что на боковой панели с пространствами больше нет пространства с заданным именем	Успешное удаление пространства по заданному имени
3.4.1	Создание приватного пространства и описанием, проверка корректности создания	Email Password Space name Privacy settings	1) Нажатие на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) Нажатие кнопки “+” (“New Space”) на боковой панели с пространствами 4) В открывшемся окне ввести название пространства 5) Нажать кнопку “Make Private” 6) Нажать “Continue” 7) В открывшемся выбрать workflow “Starter” 8) Нажать “Create Space”	Успешное создание приватного пространства

№	Название	Входные данные	Описание действий	Результат
			9) Нажать кнопку с названием пространства на панели слева 10) На открывшейся странице работы с пространством нажать “Options” (кнопка с 3 точками) 11) В появившемся окне нажать “Space settings” 12) В появившемся окне нажать “All Space settings” 13) В открывшемся окне открыть “Shared With” 14) Проверить приватно ли поле по тексту на кнопке внизу	
3.4.2	Проверка стандартных настроек приватности и описанием, проверка корректности создания	Email Password Space name	1) Нажатие на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) Нажатие кнопки “+” (“New Space”) на боковой панели с пространствами 4) В открывшемся окне ввести название пространства 5) Нажать “Continue” 6) В открывшемся выбрать workflow “Starter” 7) Нажать “Create Space” 8) Нажать кнопку с названием пространства на панели слева 9) На открывшейся странице работы с пространством нажать “Options” (кнопка с 3 точками) 10) В появившемся окне нажать “Space settings” 11) В появившемся окне нажать “All Space settings” 12) В открывшемся окне открыть “Shared With” 13) По тексту на кнопке внизу проверить, что поле не приватно	Успешное создание не приватного пространства
3.5	Создание пространства с выбором цвета и иконки и описанием, проверка корректности создания	Email Password Space name RGB color Icon name	1) Нажатие на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) Нажатие кнопки “+” (“New Space”) на боковой панели с пространствами 4) В открывшемся окне ввести название пространства 5) Нажать на кнопку под надписью “icon & name” 6) В появившемся окне выбрать цвет и иконку	Успешное создание пространства с заданным цветом и иконкой Проблема: Отсутствие прямого способа закрыть окно выбора цвета и иконки. Закрытие возможно только нажатием на

№	Название	Входные данные	Описание действий	Результат
			7) Нажать “Upload” 8) Закрыть появившееся окно 9) Нажать “Continue” 10) В открывшемся выбрать workflow “Starter” 11) Нажать “Create Space” 12) Нажать кнопку с названием пространства на панели слева 13) На открывшейся странице работы с пространством нажать “Options” (кнопка с 3 точками) 14) В появившемся окне нажать “Space settings” 15) В появившемся окне нажать “All Space settings” 16) В открывшемся окне нажать “Avatar” 17) В открывшемся окне сравнить цвет и иконку с заданными	пустое пространство страницы или через открытие другого окна, что может быть неочевидным для пользователя. При автоматизации тестирования закрыть окно без открытия “Upload” невозможно, так как оно перекрывает весь экран, делая невозможным нажатие на пустое пространство.
3.6	Создание пространства с выбором workflow и описанием, проверка корректности создания	Email Password Space name Workflow number	1) Нажатие на кнопку ‘Login’ 2) Авторизация пользователя на открывшейся странице 3) Нажатие кнопки “+” (“New Space”) на боковой панели с пространствами 4) В открывшемся окне ввести название пространства 5) Нажать “Continue” 6) Выбрать требуемый Workflow из 4 представленных 7) Нажать “Create Space” 8) Нажать кнопку с названием пространства на панели слева 9) На открывшейся странице работы с пространством проверить наличие View, которые должны были создаться при выборе Workflow	Успешное создание пространства с заданным workflow

ПРИЛОЖЕНИЕ Б

UML-ДИАГРАММА