

Программа для рисования

План разработки ПО

Software Development Plan (SDP)

Содержание

1. Описание продукта	3
2. Описание команды	4
2.1. Требуемые знания/опыт.....	4
2.2. Члены команды, их роли.....	5
3. Описание модели процесса разработки.....	6
3.1. Модель разработки.....	6
3.2. Фазы разработки	6
3.3. Ведение разработки.....	6
4. Суть проекта.....	7
5. План создания проекта.....	8
6. Контроль версий.....	10
6.1. Правила ведения репозитория	10
6.2. Оформление коммитов.....	10
7. Используемые инструменты	11

1. Описание продукта

Тип приложения: РС

Целевая аудитория: любые возраста

Продуктовое описание: Программа для рисования, по сути и основному функционалу тот же Paint. При входе в приложение появляется главное меню, где пользователь выбирает, начать новый проект или продолжить старый и начинает рисовать. По умолчанию пользователь будет использовать кисть. Появляется белый холст, и по нажатию на экран или клику мыши происходят изменения. Пользователь может менять используемый прибор между кистью, заливкой, ластиком и фигурами: кругом и квадратом, а также менять цвет закраски. После того как пользователь завершил рисование, он может либо сохранить рисунок в формате PNG, либо сохранить работу и в будущем продолжить рисовать её.

2. Описание команды

2.1. Требуемые знания/опыт

ПРОЦЕСС РАЗРАБОТКИ	ТРЕБОВАНИЯ
План разработки ПО	<ul style="list-style-type: none">• Знание моделей разработки ПО• Опыт в разработке подобных продуктов
Написание требований к разработке	<ul style="list-style-type: none">• Знание этапов разработки ПО• Полное понимание разрабатываемого продукта• Понимание технических спецификаций операционных систем
Пользовательский интерфейс(GUI)	<ul style="list-style-type: none">• Опыт проектирования GUI
Реализация	<ul style="list-style-type: none">• Знание языка Python, в рамках структурного проектирования и ООП• Знание библиотеки PIL (Python Imaging Library) для работы с изображениями
Тестирование	<ul style="list-style-type: none">• Понимание базовых принципов тестирования (Модульное тестирование)• Ручное тестирование• Опыт составления тест-планов и написания модульных-тестов

2.2. Члены команды, их роли

Члены команды	Кондаков Иван
Основная обязанность	Полная разработка приложения
Опыт	Разработка небольшого проекта на Python, знание основных принципов ООП
Знания Python	Intern

3. Описание модели процесса разработки

3.1. Модель разработки

Модель разработки: Waterfall (каскадная) - подразумевает последовательное прохождение фаз, каждая из которых должна начаться после завершения предыдущей.

3.2. Фазы разработки

Анализ и сбор требований - 1 неделя

Проектирование программной системы - 2 недели

Реализация - 2 недели

Тестирование - 1 неделя

3.3. Ведение разработки

Член команды будет разрабатывать всё самостоятельно, но при необходимости, будет взаимодействовать с преподавателем через Telegram или при очных встречах. Встречи организуются обеими сторонами при необходимости.

4. Суть проекта

Программа для рисования - программа, в которой пользователь рисует что хочет, используя имеющиеся инструменты.

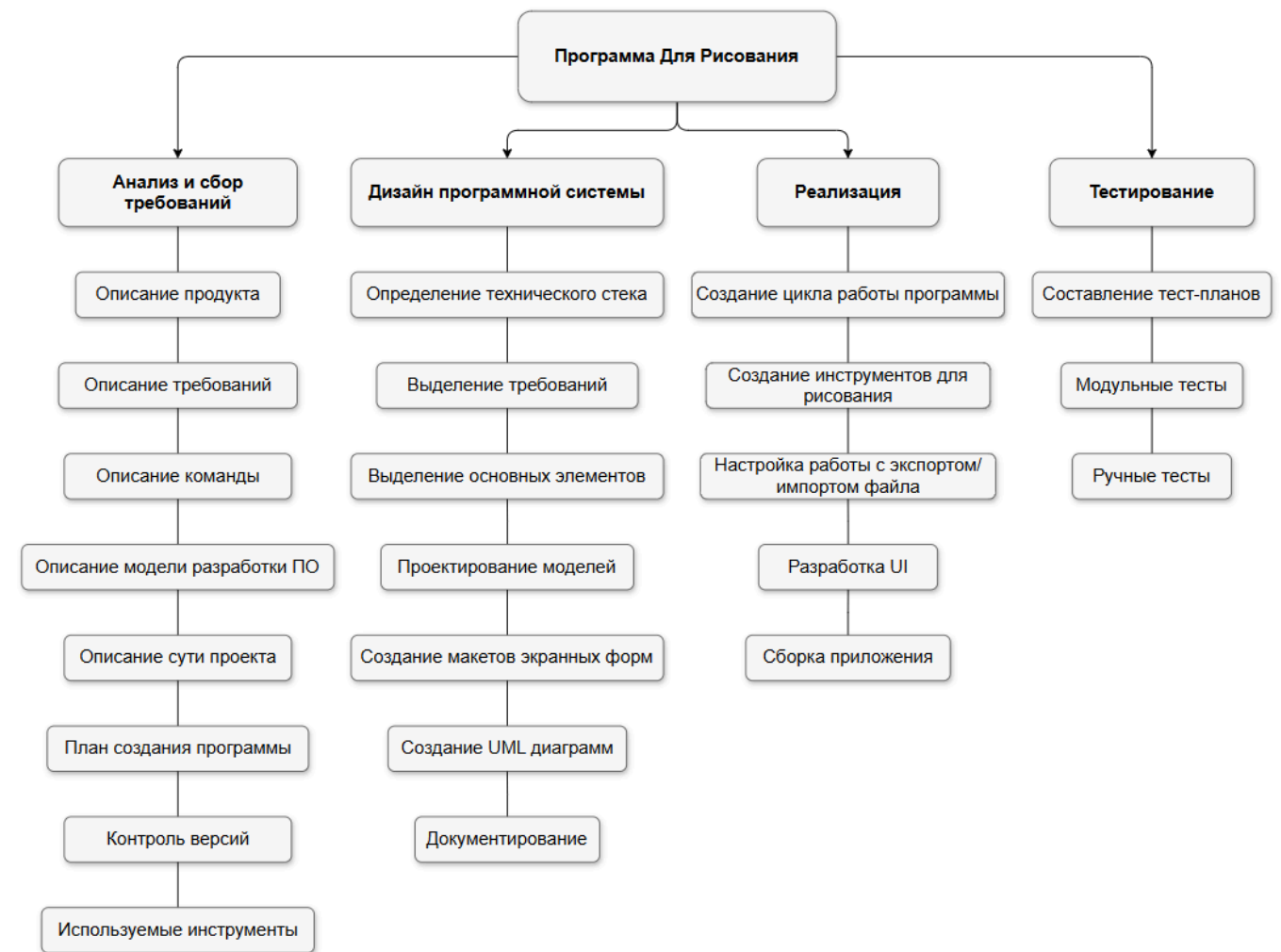
Холст: место, на котором пользователь, непосредственно, рисует
При сохранении холст .

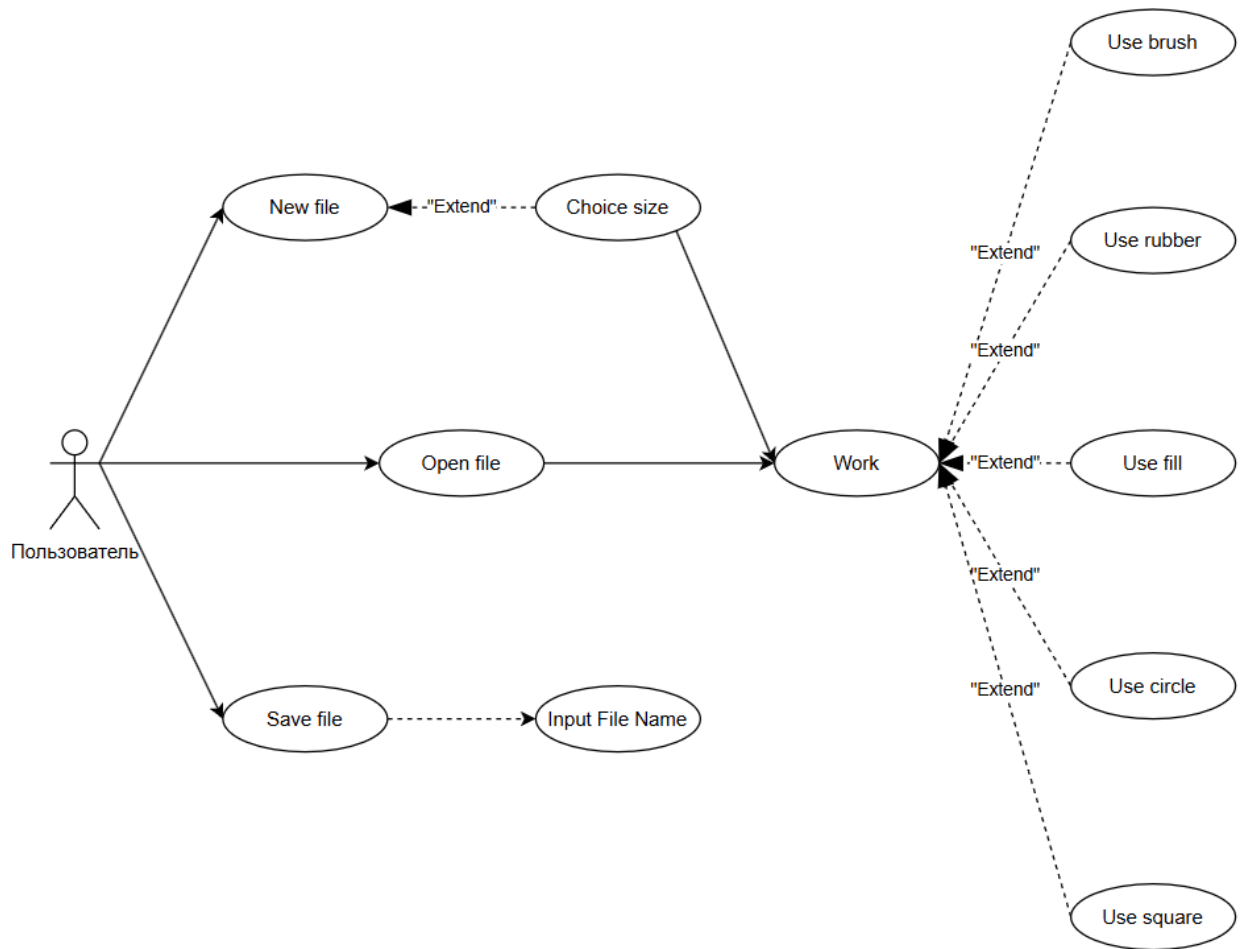
Инструменты: расположены сбоку от холста. При нажатии на выбранный инструмент им можно пользоваться.

Пользователь: самостоятельно устанавливает цель своей работы.

Управление: производится в основном с помощью мыши, а также хот-кеев.

5. План создания проекта





К концу должно быть реализовано:

- **SDP**
- **SRS**
- **Дизайн программной системы (UML-диаграммы)**
- **MVP**
 - основной цикл работы программы
 - инструменты для рисования
 - импорт и сохранение рабочего файла

6. Контроль версий

Для эффективного, удобного и простого управления разработкой проекта используется система контроля версий Git.

Ссылка на репозиторий github:

<https://github.com/kondak12/drawing-program>

6.1. Правила ведения репозитория

В репозитории имеется одна главная ветка **main**:

main - в нее вливается ветка dev, когда в ней содержится готовая версия продукта или обновления для него без каких либо багов, ошибок, недоработок и т.п.

6.2. Оформление коммитов

Коммиты пишутся следующим образом:

Тип коммита: описание

Типы коммитов:

- **feat:** добавление нового функционала
- **fix:** исправление ошибок
- **refactor:** Правки кода без исправления ошибок или добавления новых функций
- **delete:** удаление ненужного/лишнего функционала
- **test:** добавление/изменение тестового функционала
- **comment:** комментирование кода

Примеры коммитов:

- feat: add class brush
- delete: delete class brush

7. Используемые инструменты

Инструменты	Назначения
PyCharm Community	Программа, используемая для написания кода на языке Python.
Draw.io	Сайт для создания диаграмм.
Git	Система контроля версий.
Github	Сайт для ведения репозитория.
Google Docs	Написания, редактирования и просмотр документов в реальном времени.