**需求分析报告**

1. 项目目标

该模块的主要目标是提供一个能够在不同的操作系统上获取屏幕图像和颜色的跨平台的截图库。

2. 功能需求

屏幕获取：项目需要提供能够获取整个屏幕或屏幕的某个特定区域的图像的方法。

颜色获取：项目需要提供能够捕获屏幕上特定位置的颜色的方法。

跨平台：项目需要在不同的操作系统上运行，包括Windows和Linux系统。

3. 性能需求

效率：屏幕获取和颜色获取的操作需要高效执行，以最小化对用户体验的影响。

资源占用：项目需要尽可能地减少对系统资源的占用，包括内存和CPU。

4. 可维护性需求

模块化：项目的代码需要模块化，每个模块都有明确的职责，这样可以方便地添加新的功能或修改现有的功能。

兼容性：项目需要能够兼容不同的屏幕捕获技术。在Windows系统上可以使用WinAPI，而其他系统上可以使用其他技术。

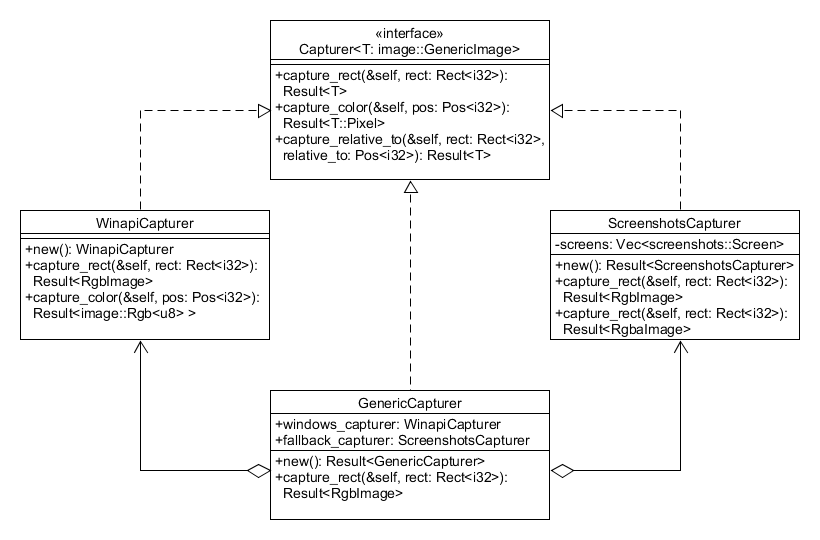
文档：项目需要提供详细的文档，包括API文档和使用指南，以帮助开发者理解和使用这个库。

5. 总结

根据以上需求，我们可以得出这是一个需要在多平台上运行，能够捕获屏幕图像和颜色，且具有良好性能和可维护性的项目。

**系统建模报告**

这个系统由以下四个模块组成：Capturer、ScreenshotsCapturer、WinapiCapturer和GenericCapturer。其中Capturer是接口，ScreenshotsCapturer、WinapiCapturer和GenericCapturer是类。每个类都定义了一种截图方式，都实现了trait Capturer，因此它们都有capture\_rect和capture\_color方法用于捕获屏幕上的图像和颜色。ScreenshotsCapturer和WinapiCapturer是GenericCapturer的组成成分，GenericCapturer会根据系统类型来决定使用（Linux）前者还是（Windows）后者。



**架构设计报告**

1. 这个系统的设计是模块化的，每个模块都有明确的职责：

Capturer：这个模块定义了trait Capturer，它是所有截图方式的基础。trait Capturer定义了capture\_rect和capture\_color方法，分别用于捕获屏幕上的图像和颜色。

ScreenshotsCapturer：这个模块定义了ScreenshotsCapturer结构体，它实现了trait Capturer。ScreenshotsCapturer结构体可以捕获屏幕上的图像和颜色，并返回RgbaImage和RgbImage类型的结果。

WinapiCapturer：这个模块定义了WinapiCapturer结构体，它实现了trait Capturer。WinapiCapturer结构体可以捕获屏幕上的图像和颜色，并返回RgbImage和Rgb<u8>类型的结果。这个模块只在Windows系统上可用。

GenericCapturer：这个模块定义了GenericCapturer结构体，它实现了trait Capturer。GenericCapturer结构体可以在不同的系统上使用不同的截图方式，在Windows系统上使用WinapiCapturer，在其他系统上使用ScreenshotsCapturer。

2. 系统使用了Rust的trait机制来定义截图方式的接口。这使得系统可以方便地添加新的截图方式实现。

3. 通用捕获器的设计：系统设计了一个通用的截图方式，它可以在不同的系统上使用不同的截图方式实现。这使得系统在不同的系统上都可以工作。