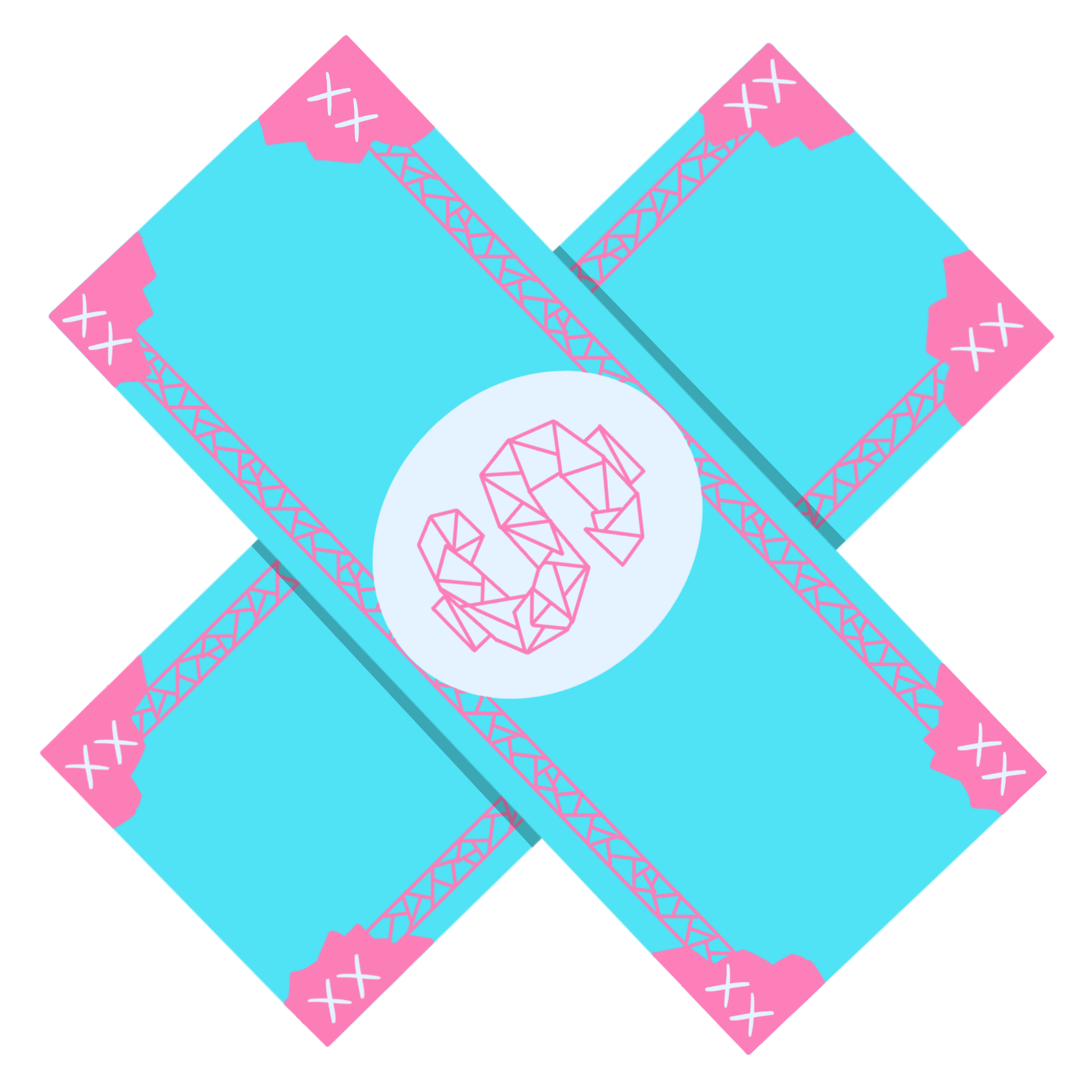
P.A.L.S.S.

Xpendit

**Software Architecture Document**



**Table of Contents**

[1. Introduction 3](#_Toc3233549)

[1.1 Purpose 3](#_Toc3233550)

[1.2 Scope 3](#_Toc3233551)

[2. Architectural Requirements 3](#_Toc3233552)

[2.1 Goals and Constraints 3](#_Toc3233553)

[2.3 Architectural Views 3](#_Toc3233554)

[3.3 Architectural Design Patterns 3](#_Toc3233555)

[3.4 Architectural Style 3](#_Toc3233556)

[3.5 Architectural Process 3](#_Toc3233557)

[3. Architectural View Decomposition 4](#_Toc3233558)

[4.1 Use-Case View 4](#_Toc3233559)

[4.2 Design View 4](#_Toc3233560)

[4.3 Process View 4](#_Toc3233561)

[4.4 Component View 4](#_Toc3233562)

[4.5 Deployment View 4](#_Toc3233563)

[4. Size and Performance 4](#_Toc3233564)

[5. Quality 4](#_Toc3233565)

# Introduction

## 1.1 Purpose

The purpose of this document is to provide a complete overview of the architecture for the application *Xpendit*. The sections of this document contain the various specific information that pertains to the architecture of the application. The main goal of the application is to provide a seamless experience to the user by using the architectural designs laid out within this document.

## 1.2 Scope

This document provides a brief overview of the architecture of the *Xpendit* application designed for Android and iOS. Xpendit is designed by P.A.L.S.S. with the intention to provide a tool to keep track of shared expenses.

# Architectural Requirements

## 2.1 Goals and Constraints

The Xpendit architecture has been developed and designed based some on the following use cases:

1. Be able to create groups with multiple members
2. To keep track shared expenses between
3. Manage a form of payment system
4. Allow the creation of user accounts
5. To allow users to see outstanding debt
6. To allow users to pay and accept any outstanding debt

The philosophy of the system is to be simple and intuitive for the user on the front end. While having a backend that can handle all the functional requirements.

## 2.3 Architectural Views

## 3.3 Architectural Design Patterns

## 3.4 Architectural Style

## 3.5 Architectural Process

# Architectural View Decomposition

## 4.1 Use-Case View

## 4.2 Design View

## 4.3 Process View

## 4.4 Component View

## 4.5 Deployment View

# Size and Performance

# Quality