TECHNICAL INSTALLATION MANUAL

STOCKFELLOW

DEVOPPS
BRIGHTBYTE ENTERPRISES

DEMO 3

Name and Surname	**Student Number**
*Tinotenda Chirozvi	22547747
Diyaana Jadwat	23637252
Dean Ramsey	22599012
Naazneen Khan	22527533
Lubabalo Tshikila	22644106

Contents

1 Introduction

This document shows the research conducted for the StockFellow project, a digital financial savings platform to provide a modern solution to the problems faced by traditional stokvels.

2 Problem

Stokvels are vital tools used for savings for over 11 million South Africans. However, traditional Stokvels face some significant challenges. These include trust issues due to lack of transparency, manual record-keeping introduces significant vulnerability to error, and delays in payments that can erode confidence. Stokvels rely heavily on social trust, and often have several risks associated with them. Our project, StockFellow, directly addresses these issues by enabling automated, transparent financial management and smart group matching.

3 Research

3.1 System

The main challenge that users of stokvels face is finding a reliable group. A core requirement for StockFellow is smart group matching, which is done by analyzing a user's financial documents and intelligently assigning users to affordability-based financial tiers. Some prospectuve users lack formal financial histories, which requires the implementation of an alternative assesment process. The project's solution is to assign those users to the first financial tier, and allow them to advance based on platform-specific behavioural data from the user. Our system measures contribution consistency by analyzing the frequency and regularity of fixed contributions over time of a user. This allows the system to upgrade users to their most suited financial tier as they progress.

3.2 Architecture

It is necessary for our application to be able to handle thousands of concurrent users at its peak transaction times. This requirement cannot reliably be met by a monolithic architecture. The microservices architecture allows for loosely-coupled services which means independent scalability, resource optimization and overall system efficiency. Our approach uses the microservices architecture to ensure our non-functional requirements are met.

Our system needs to guarantee high performance and non-blocking processes for critical services, such as payments, tier promotions and fraud detection. Event-Driven architecture is a solution for ensuring system responsiveness and latency via a message broker.

The non-functional requirements of our modern financial platform helped us land on the selection of a modular and scalable event-driven microservices architecture.

3.3 Compliance

As a financial platform operating in South Africa, we need to ensure adherence to both South Africa's POPIA, as well as the global standards set by the GDPR. According to our research, the key compliance measures for our app include:

- Ensuring transparency with users and the use of their data, while maintaining strict accountability for data handling.
- Security controls need to be fulfilled for secure identity management. Sensitive user information should be stored securely.
- Data should remain protected even in the occurrence of cross-border data transfers.

References

[1] Neumetric. "South Africa POPIA Compliance Checklist for Enterprises." *Neumetric Journal*, September 5 2025. https://www.neumetric.com/journal/south-afica-popia-compliance-checklist-2125/

[2] Katleho Mokoena, Zandri Dickason-Koekemoer, and Suné Ferreira-Schenk. "Analysing the risk tolerance levels of stokvel investors." *Cogent Social Sciences*, Vol. 7 (1): 1943890, 2021. DOI: 10.1080/23311886.2021.1943890. https://www.researchgate.net/publication/352783555_Analysing_the_risk_tolerance_levels_of_stokvel_investors

[3] Sameer Paradkar. "Designing for Uptime: A Framework for High Availability Systems." *Medium (Oolooroo)*, January 19 2025. https://medium.com/oolooroo/designing-for-uptime-a-framework-for-high-availability-systems-9d0caeb42397