Title of this Report

Author Name

August 5, 2015

Contents

Chapter 1

Introduction

this is a sample text. this is a sample text. this is a sample text. this is a sample text.

this is a sample text. this is a sample text. this is a sample text. this is a sample text.

1.1 Earlier work

this is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text.

1.2 Recent Work

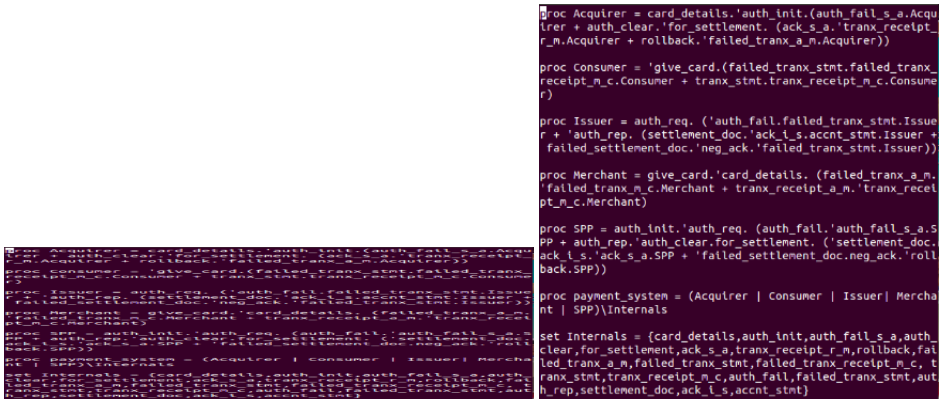
this is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text.

1.3 Survrys in this area

This is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text. Approaches discussed Section in 1.1 do not include the generic surverys.

This is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text as shown in Figure 1.1.

This is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text [3]. this is a sample text [1]. this is a sample text [2]. this is a sample text. this is a sample text. this is a sample text. this is a sample text. this is a sample text.



```
proc Acquirer = card_details.'auth_init.(auth_fail_s_a.Acquirer + auth_clear.'for_settlement. (ack_s_a.'tranx_receipt_r_m.Acquirer + rollback.'failed_tranx_a_m.Acquirer))

proc Consumer = 'give_card.(failed_tranx_stmt.failed_tranx_receipt_m_c.Consumer + tranx_stmt.tranx_receipt_m_c.Consumer)

proc Issuer = auth_req. ('auth_fail.failed_tranx_stmt.Issuer + 'auth_rep. (settlement_doc.'ack_i_s.accnt_stmt.Issuer + failed_settlement_doc.'neg_ack.'failed_tranx_stmt.Issuer))

proc Merchant = give_card.'card_details. (failed_tranx_a_m.'failed_tranx_m_c.Merchant + tranx_receipt_a_m.'tranx_receipt_m_c.Merchant)

proc SPP = auth_init.'auth_req. (auth_fail.'auth_fail_s_a.SPP + auth_rep.'auth_clear.for_settlement. ('settlement_doc.'ack_i_s.'ack_s_a.SPP + 'failed_settlement_doc.neg_ack.'rollback.SPP))

proc payment_system = (Acquirer | Consumer | Issuer| Merchant | SPP)\Internals

set Internals = {card_details,auth_init,auth_fail_s_a,auth_clear,for_settlement,ack_s_a,tranx_receipt_r_m,rollback,failed_tranx_a_m,failed_tranx_stmt,failed_tranx_receipt_m_c, tranx_stmt,tranx_receipt_m_c,auth_fail,failed_tranx_stmt,auth_rep,settlement_doc,ack_i_s,accnt_stmt}
```

Figure 1.1: A screenshot

Chapter 2

Overview

this is a sample text. this is a sample text. this is a sample text. this is a sample text.

this is a sample text. this is a sample text. this is a sample text. this is a sample text.

Table 2.1: Basic Data

Name	bf Age	**Address**
arima	22	New delhi somewhere in the lane
anok	21	TrichUr somewhere on a boat in the lake

Chapter 3

Mathematical formulation

π series by Ramanujan is as follows:

$\frac{1}{\pi} = \frac{\sqrt{8}}{9801} \times \sum_{n=0}^{\infty} \times \frac{(4n)!}{(n!)^4} \times \frac{2390n+1103}{396^{4n}}$

$\overline{True} = False$

This is important

3.1 Equations with references

$$f(x) = x^2 \tag{3.1}$$

The line defined by equation 3.1 is shown in the figure.

3.2 Equation Arrays

$$a = b \tag{3.2}$$

$$c = d \times e \tag{3.3}$$

Chapter 4

Conclusions and Future Work

These are the conclusions of my work. These are the conclusions of my work. These are the conclusions of my work. These are the conclusions of my work. These are the conclusions of my work. These are the conclusions of my work. These are the conclusions of my work. These are the conclusions of my work. These are the conclusions of my work. These are the conclusions of my work.

These are the limitations. These are the limitations. These are the limitations. These are the limitations. These are the limitations. These are the limitations. These are the limitations. These are the limitations. These are the limitations.

In future these can be done. In future these can be done. In future these can be done. In future these can be done. In future these can be done. In future these can be done. In future these can be done. In future these can be done. In future these can be done. In future these can be done.

Bibliography

[1] M. Codish, K. Marriott, and C.K. Taboch. Improving program analyses by structure untupling. *Journal of Logic Programming.*

[2] J. Hu, H.R. Wu, A. Jennings, and X. Wang. Fast and robust equalization: A case study. In *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, (SCI 2000), Florida, USA, 23-26 July 2000*, pages 398–403, FL, USA, 2000. International Institute of Informatics and Systemics.

[3] Bernd Meyer. A constraint-based framework for diagrammatic reasoning. *Applied Artificial Intelligence*, 14:327–344, 2000.