|  |
| --- |
| School of Electrical and Information Engineering  University of the Witwatersrand  [Wits EIE Logo](http://www.eie.wits.ac.za/) |
| ELEN7045 - SD Methodologies, Analysis and Design |
| The Account Presentation System |
| Submitted by:  **Group 3**  Silas Mahlangu, 699358  Livious Ndebele, 701300  Sibusiso Zwane, 517473  Boitumelo Mahlong,  Bakwanyana Thobela, 855470  Ronald Menya, |
|  |
| 6/26/2014 |

|  |
| --- |
|  |

Abstract

Contents

[1. Introduction 4](#_Toc391299795)

[2. Background 4](#_Toc391299796)

[3. Development Methodology 4](#_Toc391299797)

[4. Analysis 5](#_Toc391299798)

[4.1. Problem Context Understanding 5](#_Toc391299799)

[4.2. Specification by Example 5](#_Toc391299800)

[5. Design 6](#_Toc391299801)

[5.1. Domain Driven Design 6](#_Toc391299802)

[5.1.1 Ubiquitous Language 6](#_Toc391299803)

[5.1.2 Domain Model 6](#_Toc391299804)

[5.1.2.1 Entities and Value Objects 6](#_Toc391299805)

[5.1.2.2 Repositories 6](#_Toc391299806)

[5.1.2.3 Factories 6](#_Toc391299807)

[5.1.2.4 Domain Services 6](#_Toc391299808)

[5.1.3 Layered Architecture 6](#_Toc391299809)

[5.2. Test Driven Development 6](#_Toc391299810)

[6. Implementation 6](#_Toc391299811)

[6.1. Design Patterns 6](#_Toc391299812)

[6.2. SOLID Principle 6](#_Toc391299813)

[7. Discussion 6](#_Toc391299814)

[8. Conclusion 6](#_Toc391299815)

[References 7](#_Toc391299816)

[Appendix 8](#_Toc391299817)

# Introduction

This report discusses the solution to the Account Presentation System (APS) problem. A background to the APS problem is provided to set the stage for the rest of the problem solution discussion. The development methodology adopted is explained, and then followed by an in-depth examination of the analysis, design, and implementation of the solution.

The discussion is backed by the material that has been covered in the ELEN 7046 course [1] e.g. requirement communication techniques, Test-Driven Development (TDD), Specification By Example, Analysis and Design Patterns, Domain-Driven Design (DDD) and UML.

# Background

Computer systems have become a vital channel for delivering paperless solutions to the consumer. The Web in particular has enabled consumers to view information anytime and anywhere.

Account statements viewing is of no exception. Gone are the days when a customer only relied on postal services like the Post Office and other courier companies to deliver statements. Billing companies have realized that the need to deliver statements online gives them a competitive edge.

Moreover, amalgamation of statements from various Billing companies a customer subscribes to is an added advantage in that it provides a centralized point at which a customer can view their statements at once. The goal of the APS problem is to provide a single point of access for all of a customer’s account statements. The APS core intent is to periodically scrape customers’ account statements from their respective Electronic-Billing websites and ensure that the customers can view these statements at single focal point.

# Development Methodology

The development methodology adopted in the analysis, design and implementation of the APS solution is Agile [2]. No specific type of Agile methodology has been chosen because most require a certain set of rules to be obeyed that aren’t feasible due to the following reasons:

* Team members’ geographical distance
* Team members’ day jobs (member availability is impacted)

However the Specification by Example [3] technique has been adopted as the Agile approach to understanding the APS requirement.

# Analysis

# Problem Context Understanding

To gain a high level understanding of the problem the APS is meant to solve, the call for understanding the problem world the APS is intended to control is imperative. According to Michael Jackson [4], clearly understanding the problem world (The physical domains the machine is target for) is crucial. Figure 1 is a high level context diagram that illustrates the relationship between the machine (APS) and the physical domains that have been identified.

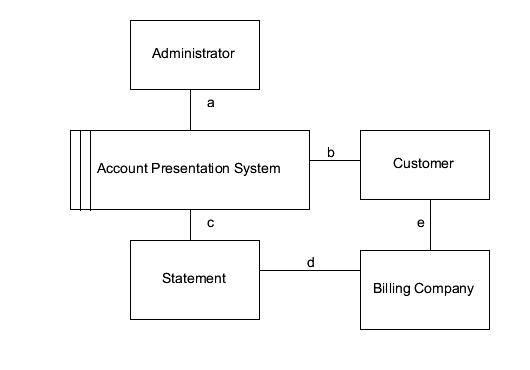


Figure 1: Relationship between the APS and the problem world

The relationships or shared phenomena [4] between the domains in figure 1 are influential in the outcome of the APS specification. Below is high level description of the shared phenomena.

1. An administrator schedules statement scraping.
2. A customer registers with the APS, configures statement viewing, and views statements.
3. The APS scrapes/retrieves a statement.
4. A statement is from billing company.
5. A customer has an account with a billing company.

# Specification by Example

* Why?
* How?

# Design

# Domain Driven Design

# Ubiquitous Language

# Domain Model

# Entities and Value Objects

# Repositories

# Factories

# Domain Services

# Layered Architecture

# Test Driven Development

# Implementation

# Design Patterns

# SOLID Principle

# Discussion

# Conclusion

# References

[1] SP Levitt, J Lewis

[2]

[3]

[4] M.Jackson, Problem Frames: Analyzing and structuring software development problems, Addison-Wesley, 2001

# Appendix