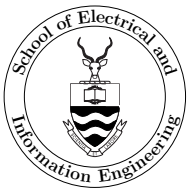

UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG
School of Electrical and Information Engineering
SD Methodologies, Analysis and Design

Course Project — Account Presentation System

1 Introduction

Many companies offer their customers the option of viewing their account statements online. These companies range from retail stores, such as Edgars, to utilities such as City of Joburg. A customer who wishes to understand their financial situation at any point in time has to visit multiple websites and manually aggregate account information from different companies.

You are to design and construct an Account Presentation System (APS) to alleviate this situation. This system will provide a single point of access for all of a customer's account statements.

2 Website Scraping

A key component of this system is the *website scraper*. The scraper is a third party product which is capable of controlling a web browser and parsing html pages in order to obtain information. Scraping scripts detail the navigation process and which fields are to be scraped. The scraper works in the following manner: given a starting point `url` and user credentials, it will navigate to the required page(s) and extract specified text fields and associated values. For example, text in one column may be associated with a number in an adjacent column to its right. These extracted text-value pairs are stored in an XML file. The XML file also contains the date and time that the scraping was performed as shown in Figure **??**.

The scraper's task is to create an XML file storing the data specified by the scraping script. It does not perform any calculations related to the data.

2.1 Error Conditions Detected by the Scraper

A number of possible error conditions are detected by the scraper. These errors are reported via the XML file. When an error occurs the XML file will contain only the base-url, date and time of the scraping attempt and the relevant error code, if the source of the error can be detected.

2.1.1 Invalid Credentials

During a scrape, the credentials that have been supplied by the customer may be found to be invalid. This results in the user being presented with an invalid login page which is detected by the scraper.

This may be due to the credentials having been entered incorrectly by the user on account activation or due to the credentials having been changed by the user subsequent to them having been supplied to APS.

```
<scrape-session>
<base-url>www.telkom.co.za</base-url>
<date>10/01/2008</date>
<time>13:50:00</time>
<datapair id="001">
    <text>Account no</text>
    <value>53844946068883</value>
</datapair>
<datapair id="002">
    <text>Service ref</text>
    <value>0117838898</value>
</datapair>
<datapair id="003">
    <text>Previous Invoice</text>
    <value>R512.22</value>
</datapair>
<datapair id="004">
    <text>Payment</text>
    <value>R513.00</value>
</datapair>
<datapair id="005">
    <text>Opening Balance</text>
    <value>R0.78</value>
</datapair>

...

</scrape-session>
```

Figure 1: Example XML file produced by the scraper.

If the credentials are found to be invalid, the service should notify the user and request that the user supply their updated credentials before the scrape is attempted again. The user's account status is changed from "active" to "inactive". Once a user has updated their details, their account status is changed to "trying".

2.1.2 Customer Not Signed Up for e-Billing

Many companies' websites allow their customers to register to use the website, but do not immediately give them access to their account information. The customer must go through a separate sign-up process in order to "activate e-billing". The customer's credentials may be perfectly valid, but the scraper will be unable to access the account statements on behalf of the customer.

If the customer has not yet signed up to e-billing and they attempt to navigate to their account information, they will typically be presented with an instruction page or be directed to the e-billing signup process. It is assumed that this scenario is standard for some billing companies and that the scraper can be programmed to identify it by parsing the response pages for expected data. The scraper will then return the relevant error code.

If the customer has not yet signed up to e-billing, APS should notify the customer and request that the customer complete the signup process before the scrape is attempted again.

2.1.3 Action Required by Billing Company's Website

There is the possibility upon logging in to the billing company's site that the customer is required to take some action before being able to navigate to their account information.

This scenario will typically occur after the customer has been using the service for some time, and some condition occurs that necessitates the billing company requiring some interaction with the customer. For example, the company may wish to update the profile information that it holds for the customer (such as the customer's email address), or the customer may need to accept a new set of terms and conditions for the use of the company's website.

This type of error differs from the previous one in that it is unexpected. Additionally, the scraper cannot be programmed to take an action on the customer's behalf in these situations. For instance, APS cannot accept terms and conditions on behalf of the customer. Instead, the service needs to notify the customer that some action needs to be taken. In other words, the customer is given the opportunity to resolve this condition before the scrape is attempted again.

2.1.4 Billing Company's Site Down

When the billing company's site is down for scheduled or unscheduled maintenance, a page is typically displayed to notify the customer of the unavailability of the service and possibly provide details of when service will be restored. If the scraper is able to identify this page, then the site down condition can be handled accordingly - generally by delaying for a period of time before attempting the scrape again. A sophisticated implementation could potentially parse this page to determine the service restoration timeframe if this information is provided. These details could then be used to influence when the scraper next attempts the scrape.

Note that periods of scheduled maintenance are best accounted for by respecting the billing company's known maintenance time periods by using the "site closed window" functionality described in Section **??**.

2.1.5 Error Page Encountered

The company's site may sometimes display a specific error page which indicates that a certain area of service functionality is temporarily unavailable. This may be intermittent (for example, based on load) or it might be as a result of service maintenance that the billing company is performing.

This condition is similar to the "site down" scenario described in Section **??**, but the distinction is that in this case the company's site as a whole is not unavailable, so the scraper may be able to log into the site, but not access the account content. The action to be taken upon encountering this condition is similar to that taken when the company's site is down i.e. delaying for a period of time before attempting the scrape again. However, a different delay might be configured since this condition will generally be more transient and may be resolved more quickly than the "site down" condition.

2.1.6 Broken Script

This is a "catch-all" error condition to cater for unanticipated behaviour that is encountered when scraping the billing company's site. There are various reasons that such a condition may be encountered. Two of the most common are that the site has changed (generally as a result of regular service maintenance or enhancement work by the company), or a data condition or statement variation is encountered that has not been seen previously. These conditions are described in more detail below. Typically, they will necessitate changes being made to the scraping script; the script will need to be analysed by a developer to determine the nature of the failure and to fix the script so that it functions properly.

Site Changed
A billing company's site changes may alter the content and position of data elements or of text anchors that the scraper uses to locate them. The site navigation might also change such that the page flows are different to those that the scraper expects. While the scraper script is constructed in a non-brittle way that is forgiving of relatively minor changes to a billing company's site, it is not possible that the scraper script will be able to cater for all possible changes. However, the script should reliably recognise when the site is not behaving in an expected way and generate a "broken script" condition.

Encountered Unhandled Data Condition
There is often a fair amount of variation in the way account statements are presented on the billing company's website, depending on the type of account and the status of the account. For example, statements that show a zero or credit balance may be represented differently to statements that have a positive amount due, or the format of a re-issued statement or reminder might be slightly different to the original statement.

One objective of the analysis process that is undertaken before the scraper script for a particular billing company's website is built, is to identify the different variations that could occur so that the scraper script can deal with these conditions appropriately. However, it is possible that a running script encounters an unlikely variation that has not been catered for. Under these circumstances, the script generates a "broken script" error condition so that the failure can be analysed and the script enhanced to enable it to deal with that variation.

3 APS Requirements

3.1 Credentials Management

In order to allow the scraper to gain access to the billing company's site, the customer will need to supply their credentials. These credentials will need to be stored, handled and managed securely.

3.1.1 Credential Types

Two types of credentials have been identified:

1. Identification fields – These fields serve to identify the customer to the billing company. They are typically entered by the customer in plain text and will be presented back to the customer in plain text for confirmation or if the customer wishes to update them. Examples are username, first/last name, email address, account number, telephone number, or some other specific identifier that is issued to the customer purely for the purpose of accessing the billing company's website.
2. Authentication fields – These fields (which are also referred to as security fields) are typically entered by the customer without the input being echoed to the display (the entered value is obscured by asterisks). These fields contain sensitive data that the customer uses to demonstrate his/her proof of identity and hence authority to access account information on the billing company's website. Examples include password, PIN number, security code etc. These fields are typically entered twice for confirmation and never presented back to the customer.

The data contained in identification fields is not particularly sensitive and can be held in plain text. This allows them to be presented back to the customer without the need for decryption. Authentication fields need to be stored encrypted. Authentication credentials should be decrypted only immediately before they are used and the plain text should be overwritten and destroyed immediately when it is no longer required. Note that certain fields, such as date of birth or credit card number, that are entered in plain text might still be deemed sensitive and need to be encrypted. In these cases, the field contents will need to be decrypted if the field is to be presented back to the customer.

APS should cater for up to 5 plain text and 5 encrypted fields per account. The system should allow for a maximum field length of 100 characters.

3.2 Data to be Scraped

The range of data fields that are to be scraped will vary by statement type and will be programmed into the relevant scraping script. However, it is desirable for common fields that are scraped across a number of companies to be presented to the customer in a consistent way. A number of standard terms have been defined, and text data that is scraped should conform to these terms where possible. During analysis for a particular billing company, a mapping needs to be performed between the fields that are available on the company's website and the respective terms that will be used when presenting statement data from that company.

3.2.1 Common Data

Certain data fields have been identified as being common to all statement types. These may be directly present, or can be derived.

Common Fields
Account number
Account holder name
Statement date
Statement number
Statement month
Total due
Due date
Opening balance
Closing balance
Payment received
New charges
Deductions
Discount
VAT amount

Table 1: Fields common to all statements.

3.2.2 Statement-Specific Data

A number of specific fields have been defined for particular statement types, namely for statements from:

- Municipalities
- Credit Card Providers
- Telecommunication Service Providers

Municipalities	Credit Card Providers	Telco Service Providers
Instalment notice	Card type	Telephone number
Electricity used	Interest rate	Service charges
Electricity charges	Credit limit	Call charges
Gas used	Credit available	Total number of calls
Gas charges	Minimum amount due	Total call duration
Water used		
Water charges		
Sewerage charges		
Refuse charges		

Table 2: Fields specific to certain types of statements.

The fields that have been identified for each statement type are detailed in Table **??**. It is expected that additional statement types will identified in future.

3.2.3 Numeric Data

Numerical data is scraped as text. There is a need to convert this data to a numerical representation in order to perform integrity checks (Section **??**). Numerical data conversion has to take into account different currency formats, for example, negative amounts may be represented in different ways.

There also a need to present this data consistently to users of APS (as with text data fields). In some cases, transformations will need to be applied to scraped numerical data in order to ensure a consistent presentation across statements. This may involve changing currency formats, or the sign of numeric values, or even amalgamating values into a single field.

3.3 Data Integrity Checks

Data that is scraped from a company website will be subjected to a series of integrity checks to ensure its accuracy. The scope of these checks is currently intra-statement (checking that data is consistent within a particular statement), although inter-statement checks (checks that span across statements) may be contemplated in future.

For a particular billing company, the range of checks that can be performed will vary depending on the data that is available and the way in which that data is presented on the billing company's website.

3.3.1 Standard Integrity Checks

A number of standard integrity checks have been identified. These checks may not be able to be applied in all cases, but they represent a common set of checks that should be considered for each billing company. The goal should be to perform as large a subset of these checks as possible.

Addition Checking
For most statements the following calculation can be done:

	Opening Balance
–	Payment Received
+	New Charges
–	Discount
–	Deductions
	Total Due

VAT Calculations
Many bills present a Value Added Tax (VAT) amount. In addition, the VAT rate may be present or can be assumed as a fixed percentage value (currently 14%). The VAT amount can be re-calculated using the appropriate charges amount and the VAT rate.

Correlation of Duplicate Data
Scraped field values sometimes appear in more than one place on the billing company's website. If all available instances of a particular field are scraped, a comparison can be done to ensure that the scraped values are the same. This increases the confidence level that the data has been scraped correctly since the probability is low that more than one instance of a field is scraped incorrectly in exactly the same way.

3.3.2 Custom Integrity Checks

In addition to the standard set of checks, a further set of specialised (typically billing company specific) checks may be able to be identified for a particular billing company based on the available data. In general, the more data integrity checks that can be performed on the data that is scraped, the greater the level of confidence in the accuracy of the scraped data. Therefore

any opportunities that a particular billing company site provides should be exploited. The account presentation system must cater for the addition of custom company-specific integrity checks that can be performed on web-scraped data.

3.4 Scraper Scheduling

3.4.1 Site Open and Closed Windows

It is important to be able to configure periods of time during which to avoid scraping a company's site. This may be required for the following reasons:

- To avoid a known maintenance window for the company's website (during which time the site is unavailable so scraping attempts would be futile).
- To avoid scraping during known peak demand periods (either because there is a greater chance of the scrape failing due to timeouts, or in order not to impose additional load on the company's servers during these times).

3.4.2 Statement Life-Cycle Modelling

To avoid unnecessary scraping operations, it is important to model the typical billing cycle for each company, and only scrape when a new statement is expected. The following parameters need to be maintained for each company:

- Number of days per billing cycle
- Lead-time interval — number of days prior to start of billing cycle to begin scraping
- Retry interval — interval before retry following unsuccessful scrape.

When a new statement is returned successfully, the next scrape timestamp is calculated as:

current statement date + cycle days - lead time

The lead time is initialised to zero for all companies. It may be possible to automate the setting of the billing cycle days by keeping statistics on actual "new statement found" returns. This is unlikely to be implemented in Phase I but may be a future requirement.

Naturally, it is important to distinguish between the new statement and the previous month's statement for a particular account. This can be achieved by comparing the statement date and total with the previous month's statement data.

3.4.3 Retries Following Error Conditions

Rules need to be defined for when to re-scrape an account following an unsuccessful attempt. Different retry intervals need to be specified for errors originating from the scraper (Section **??**), for errors originating from within APS (Section **??**), and for when no new statement is found. These retry intervals need to be configurable per company.

3.5 Scraper Load Management

3.5.1 Concurrent Scraping Operations

In order to keep from overloading a site with scrape traffic, APS must support configuration of how many concurrent scrapes to allow for a billing company.

3.5.2 Traffic Shaping

There may be a need in the future to support variable concurrent scrape limits by time of day in order to avoid peak demand periods. A concurrent scrape limit of zero during a particular time period is tantamount to imposing a site closed window (refer to Section **??**). Variable non-zero limits allow scraping activity to be reduced at certain times of the day in order to prevent adversely impacting a company's systems.

Account activation requests need to be prioritised over statement scraping requests, so as not to unacceptably delay activations as a result of a high scraping load.

3.6 Error Handling

A number of possible error conditions need to be detected and handled by APS, in addition, to the errors generated by the scraper (Section **??**). The APS service must handle each of these appropriately. The overriding principle is to be conservative to prevent presenting erroneous data to the customer.

3.6.1 Incorrect Account

During the account activation process, the customer will typically supply both their account number and a set of credentials for accessing the billing company's website (these details are required by the scraper). In addition to being supplied by the customer during activation, the account number is often available on the billing company's website as a data field that can be scraped. A cross-check can be done to ensure that the account number that the customer has supplied matches the account number reflected on the site. If this check fails, the activation should be rejected and appropriate messaging should be sent back to the customer to inform them that this has occurred.

3.6.2 Data Fails Integrity Checking

If any of the data integrity checks described in Section **??** fails, APS should generate this error condition. The action taken is similar to that described for the "broken script" condition in Section **??** – namely that some manual intervention is required by the APS developers to analyse the reason for the failure and potentially enhance the scraper script in order to account for a previously unanticipated condition.

3.6.3 Data as Presented Cannot be Integrity Checked

Occasionally the scraper may encounter data on the billing company's website that is actually presented incorrectly or inconsistently by the billing company. The scraper might be functioning correctly but the erroneous nature of the scraped data means that the data fails integrity checking.

In some cases the integrity checking by APS can be enhanced so that it is able to correct certain types of error. For example, certain types of rounding error can be corrected by slightly relaxing the integrity check that is performed. For example, if a calculated amount is within one cent of the amount presented by the billing company, then the data may be accepted as correct.

If it is not possible to correct the error, then the condition may need to be added to an exception list which acts as an override for the integrity check that is failing.

4 Web Front End

The front end for APS is to be kept intentionally simple for the first phase. Customer's account statements are to be presented on static web pages, and simple forms can be used for gathering the customer's credentials and any other information that is required. Customers will be required to securely log on to the APS site in order to view their account information.

5 Logging/Data Storage

A comprehensive audit trail needs to be created for the system in order to resolve customer disputes and to facilitate the correction of errors. Logging or storage of the following data is required:

- The date, time and `url` of all system scrapes.
- For successful scrapes, the scraped data *as well as* the transformed data that is presented to the customer.
- For unsuccessful scrapes, any errors that are reported.

6 System Scope and Assumptions

Secure communication over the internet as well as the strong encryption of customer credentials are vital parts of APS but for the purposes of this project you may ignore/relax the security concerns.

As mentioned the web scraper is an existing third party product. You may assume that the scraping scripts that the scraper utilises have already been created by other members of the APS development team for a number of billing companies.

The APS service is a commercial offering. The revenue model and the billing of APS customers for the use of the service is beyond the scope of this project.

7 Assessment

7.1 General

- The project brief is intentionally vague and possibly contradictory in some areas. You are required to make *reasonable* assumptions about various aspects of the system behaviour.
- The focus of this project is on the *domain layer* of the system. A large part of your analysis and design effort should concentrate on this layer. You need not cover the entire design space at a detailed level. Specifically, you should minimise the effort that you put into handling secondary concerns (from the point of view of this project) such as security, user interface design, and so on.
- You are required to apply the material that you are covering in this course to this project, including, requirement communication techniques, Test-Driven Development, SOLID Principles, Analysis and Design Patterns, Dependency Injection, Domain-Driven Design and UML.
- You are required to create a prototype which demonstrates the key design decisions that you have made. This can be done in any mainstream object-oriented language. Remember,

you are creating a proof-of-concept prototype to illustrate your particular design approach — not a full-fledged application. Each group member must be responsible for creating a *distinct part of the prototype* involving the *application's domain layer*.

7.2 Assessment Form

This course project contributes 40% towards the overall course mark. The marking grid and assessment guide that accompanies this brief will be used to assess the project.

In order to map the assessed outcomes to an overall percentage, an algorithm is used. The weighting factors used in the algorithm are summarised in the tables below. Note that the specific outcomes within each component have equal weightings. There is one overriding rule: if any outcome category is rated as *Unacceptable*, then the mark for that component is capped at 40%; however, the mark can be lower than 40%.

Project Component Weights	
Group Report	0.35
Individual Report	0.50
Presentation and Demo	0.15

Outcome Ratings	
Unacceptable	0.00
Poor	0.20
Acceptable	0.55
Good	0.70
Excellent	0.95

7.3 Group Self-Assessment

Each group of n students is allocated $5n$ percent in terms of discretionary marks. For instance, a group of five is allocated 25 percent. It is up to the group to determine how to divide this. The discretionary marks may be evenly split (5% each) if it is felt that all group members contributed equally to the project. If this is not the case, the group can acknowledge members who have contributed more than others by granting them a larger share of the marks. The maximum that any one group member may be awarded is 10%. The discretionary percentage for each group member is added to that member's overall percentage to determine the final mark for the project. An example of this, for a group of four, is shown in the following table.

Group Member	Overall %	Discretionary %	Final %
A	70	8	78
B	62	5	67
C	65	5	70
D	56	2	58

In order for the discretionary marks to be granted all group members have to agree on how the marks are apportioned and sign to this effect on a declaration form (refer to section **??**). Note that this process is not intended to account for gross differences in effort. If you feel that a group member is not pulling their weight then try to resolve this internally. If this is not possible then discuss the issue with the lecturer.

If a student receives a rating of *Unacceptable* for any outcome then that student's discretionary marks are forfeited.

8 Deliverables

Group Report

Each group will be required to submit one concise project report (no longer than 15 typed pages) that includes at least:

- A discussion of the development methodology adopted by the group;
- A complete analysis of the problem domain and the requirements;
- A *high-level* or architectural view of the design and the areas of the design that were prototyped;
- The source code of the prototype on a CD.

Preparation of this report will be a group effort.

Individual Report

Each member of the group will also have to submit an individual report (no longer than 10 typed pages, excluding appendices) that focuses *in detail* on his/her unique contribution to the system design. This report must not simply repeat sections from the group report. It should minimally include:

- a brief but clear explanation of the role of the member in the group and the specific area of the project that they were responsible for;
- the analysis relevant to the part of the domain layer being modelled;
- the key design decisions in this regard;
- the implementation of the relevant parts of the prototype and the issues raised in prototyping;
- the modifiability and reusability of the modules designed;
- suggested refactorings to improve the current design and/or alternative designs and the tradeoffs involved.

Presentation

Each group will be required to present their project, and demo their prototype, to the class in no longer than 20 minutes. All group members are required to present. This will be followed by 5 minutes of questioning/critique.

Declaration

A page declaring the discretionary percentage split among the group members must be submitted and signed by all group members. *If the declaration is not submitted or not signed by every member then no discretionary marks will be awarded.*

The completed documentation and presentations must be concise and of a high standard, in accordance with accepted guidelines for report writing and presentation delivery that apply within the School of Electrical and Information Engineering.

8.1 Submission Deadline

Refer to the course homepage for the project deadline. Note that the presentations are due prior to the documentation.

8.2 Plagiarism

All instances of plagiarism will be severely dealt with. No two groups or group members may have identical or overly similar deliverables.

9 Acknowledgements

Thanks to Warren Blumenow for providing the original specification on which this project is based.