Sala: Sales Analysis, Prediction and Promotion System

M. Z. Mohamed
Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka
maryamzi.12@cse.mrt.ac.lk

Abstract— **Clothing retailers notice variations in buying patterns of customers with time. These variations depend on factors such as the numerous brands and items being available and the time of the year. The temporal variations tend to show a certain trend across years. Sala, a Sales Analysis, Prediction and Promotion System consists of a standalone desktop application, developed in Java, which allows the management to identify what sells more (trends) and predict future sales enabling the client to increase sales quantities and profitability. Along with this system, an Android mobile application was developed, which could be used to promote these specific items of clothing, also increasing the quantity sold. The analysis provided graphically makes it easier for the management to analyse past sales and the prediction of future sales, done using least square regression, enables them to order according to possible future demand. Furthermore the Android application promoting only the items predicted to be popular makes it easier to spread information among customers with an uncluttered user friendly app. Sala is a tool enabling management to identify and analyze past and future sales and approach customers more effectively ultimately increasing sales and profitability of the client enabling the client to succeed in a competitive environment.**

Keywords—Clothing; Sales Predictor; Sales Analyser; Sales Promoter

I. Introduction

Customer buying patterns vary with the wide range of brands and items of clothing available. Ordering items for sale without analyzing customer preferences leads to unsold inventory and cause losses. Customer buying patterns also tend to vary throughout the year but they tend to show trends across years. For example buying tends to increase during festive seasons while items preferred by customers also tend to change with factors such as weather. Ordering items without considering these changes/trends would cost the client the opportunity to increase sales. Also current mobile applications available to customers from retailers are not very user friendly due to the large number of items included, which cluster the app. The search functions may not be satisfactory as the customer may not be able to specify everything he/she wants.

The motivation behind the development of the selected system was to provide management with information required to make decisions by providing analysis and comparisons of sales for different brands/items of clothing. It was also aimed to help them identify what should be ordered in order to increase sales by analyzing past patterns and predicting future demand according to the period of the year. Furthermore it was aimed to provide users with a user friendly mobile application containing only the items that would be in demand, reducing clustering, making it easier for the customer to identify items in stock and choose.

The main purpose of the system was to provide management with a mechanism to analyse past sales and predict future sales in order to optimize purchases, i.e. - order larger quantities of the items predicted to be in demand. Also the system was intended to provide a mechanism to promote the items predicted to be in demand effectively via an Android app.

This paper, in Section II, identifies the current literature available on similar systems and alternate approaches. Section III describes the functional and non-functional requirements and presents the architecture of the system. Section IV describes the implementation of the system and Section V provides an overview of the testing and analysis of the system. Section VI provides a conclusion and possible avenues for future development of the system.

II. Literature Review

Demand for clothing items changes with time, irrespective of gender. However the main factors influencing buying decisions tend to be festive occasions and changes in the weather/seasons. With major festivals and holiday seasons such as the New Year, Ramadan and Christmas significant changes in the demand for certain items of clothing can be noticed. Also changes in weather such as the rainy season cause changes to the items of clothing in demand. Although Sri Lanka does not experience major changes in the climate with seasons, in countries which experience drastic season changes, major demand changes can be identified.

For example, with winter arriving, the demand for warm clothes increases rapidly encouraging shop owners to purchase and store more warm clothing for sale [1]. Although the lifecycle of clothing items is usually short, the time taken for production is comparatively lengthier and thus orders placed suddenly may not be accommodated [2]. Therefore a proper analysis of past data to predict future sales accurately would enable management to place orders well in advance to ensure sufficient stocks of popular items are available at a given time. Also if items are purchased for sale without considering possible demand it could result in unsold inventory which would cause not only losses in sales to the company but also increase stock holding costs due to holding this unsold inventory.

Prediction of future demand in the fashion industry is a challenge, compared to other industries, mainly due to the volatility in customer preferences when it comes to fashion choices [3]. A wide range of related work is available in literature on sales prediction systems for the fashion industry [3], [4]. Huang et al. have stated [4] that the nearly 200 prediction methods identified could be classified as subjective or objective where subjective methods are based on the experience of the experts making judgements and objective methods rely on models built using data. While subjective models such as the Delphi method and the brainstorm method are highly flexible, accuracy of the predictions depend on the expert, while objective methods although not flexible are reusable. On a broad scope in predicting demand of clothing items several statistical objective techniques based on time series models such as ARIMA and SARIMA and other techniques based on a Bayesian approach have been used [3]. Considering several potential drawbacks in statistical methods, since of late, forecasting techniques incorporating an Artificial Intelligence component have been suggested to predict demand more accurately [3]. Basic statistical methods work only with certain data sets [4] while methods utilizing data mining, which help develop general models, are applicable to wide ranges of data. Hybrid models which combine statistical models and AI to make use of the advantages of each component individually have also been suggested [3].

However in Sala, prediction is based on Least Square Regression which aims to fit the best fit line considering past sales data to accurately predict future sales. Past sales data are used in calculating the gradient and intercept of the line of best fit for a particular brand's sales for a particular item of clothing. The analysis helps management to identify sales in terms of quantity sold rather than the revenue/ profit generated in order to identify exactly what sells more. This Android mobile app is focussed to reach more customers as Android is one of the mobile operating system with a large market share.

III. System Models

A. System Requirement

The main functional requirements of the system were that the system should enable analysis, prediction and promotion of sales. The desktop application developed would provide the management with the capability of analyzing and predicting sales and these predictions would be used to decide what items are advertised to the customers via the mobile app. Customers would use the mobile application to obtain details regarding the items predicted to be in demand. Depending on the security level of the user, he would be able to analyze and compare sales by viewing sales by year, month, brand and item. A user would be able to predict future sales for all items or a single item by year/month by selecting a future time period. Also in promotion the user would be able to select the period of time and number of items to be promoted to the users of the mobile app, which would in turn be displayed via the app.

The main non-functional requirement of the system was security as highly confidential sales data of the retail outlet could be accessed via the system. This information getting into wrong hands could lead to disadvantages to the company and thus it is vital that proper security is in place and only those authorized have access to sales data, future predictions and can publish popular items. Usability was also important and it had to be ensured that even new users can get used to using the system within a short period of time by minimizing user input. Additionally the reliability of the system was also expected to be high especially with regard to the predictions as incorrect predictions could lead to losses to the outlet.

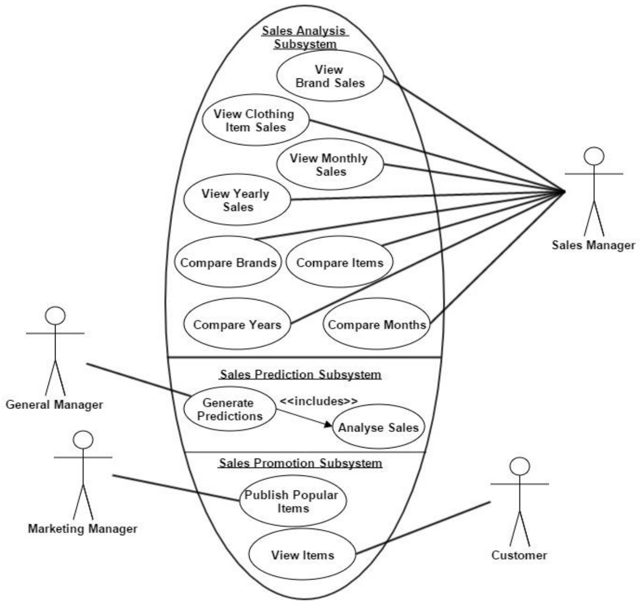


Fig. 1. Use Case Diagram.

As shown in Fig. 1, the use cases are identified with respect to the three subsystems of the system namely **Analysis**, **Prediction** and **Promotion** where promotion includes the android App too. In the analysis subsystem a **Sales Manager** would be able to analyse past sales data by viewing data relevant to brands, items, months and years and furthermore comparing past sales data between two distinct brands, items, months or years. In the prediction sub system a **General Manager** would be able to predict future sales by specifying a month/year for which predictions would be generated either for all items or a specific item as the manager prefers. Prediction generation would include an analysis of past sales. In the promotion subsystem a **Marketing Manager** would be able to publish the most popular items for a specific month by specifying the month and the number of items to be promoted. A customer using the android mobile app on his phone would use the app to view the most popular items as published above.

B. System Design

The architecture of the system could be identified as being classified into three main layers namely the **View Layer** containing the interfaces of the system to the user, for both the desktop application and the mobile app, the **Business Logic Layer,** which includes all the basic logic handling classes of the system including calculations and Handlers interfacing between the core logic, view layer and the data access layer and the **Data Access Layer,** which contains classes that performs the data access functions to retrieve and provide necessary data from the database.

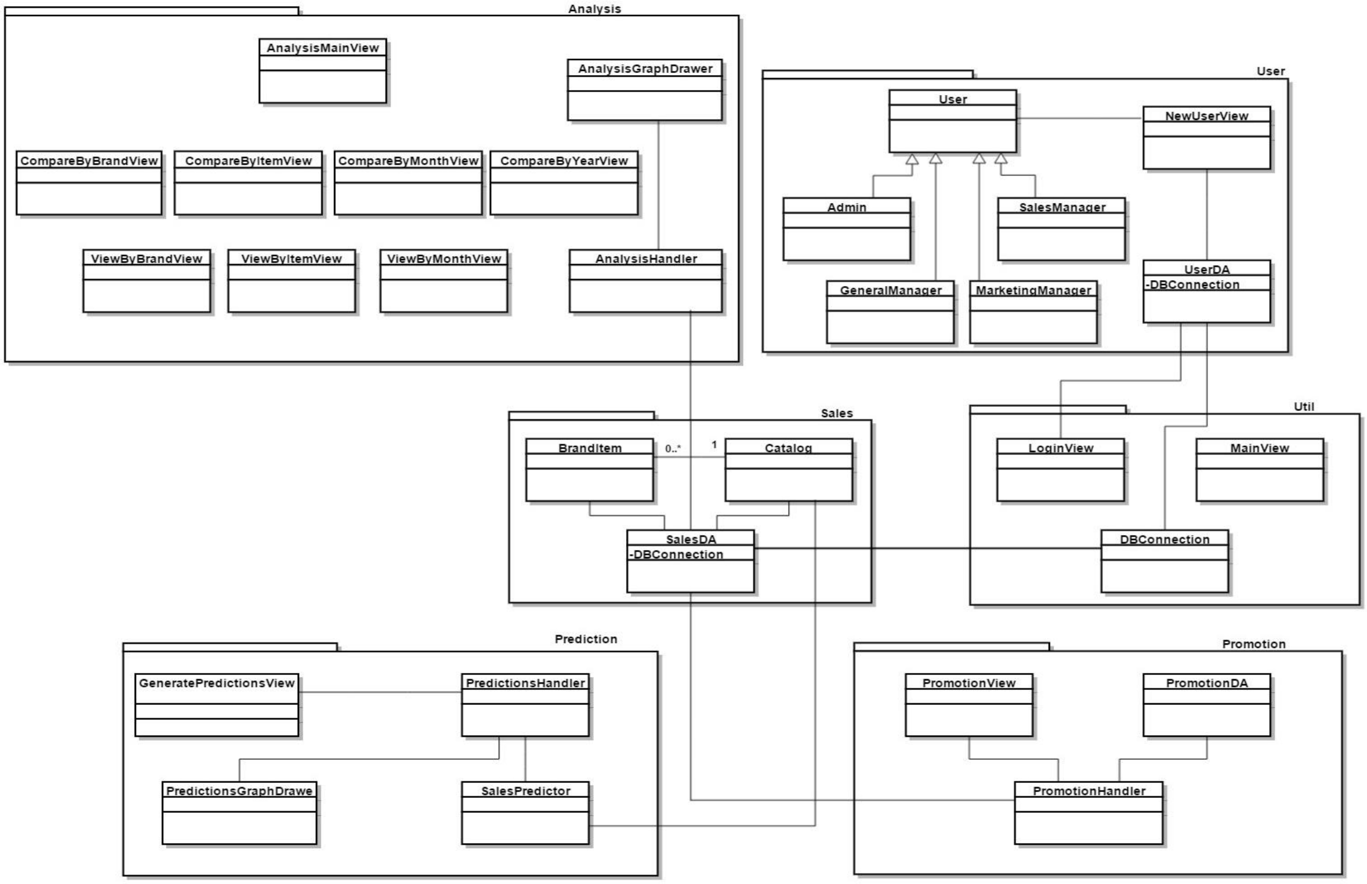


Fig. 2. Class Diagram

The system developed is classified into seven basic packages: User, Util, Sales, Analysis, Promotion, Prediction and Test. As shown in Fig. 2, the classes in the **Sales** package include the **BrandItem** and **Catalog**. They are used to maintain sales data for each item where **Catalog** includes all the items and sales details. The **SalesDA** is used to retrieve data from the database and is used by several other packages in the system including analysis, prediction and promotion. The **Util** package consists of the classes that deal with the initial functionality and the database connection. The **DBConnection** class is used by the data access classes in several other packages. The **User** package includes classes representing different users and they are instantiated whenever a user logs in and stores the security level of the user to grant permissions across the system. The **UserDA** class is used to read user data and to authenticate users login. The **Analysis** package includes the classes for views, graph drawers and handlers which deal with providing the analysis functionality. According to the analysis requirement of the user the analysis handler reads data from the database using the **SalesDA** class and calls the relevant method in the **GraphDrawer** class to present the data graphically. The **Prediction** package includes classes for predicting the sales including the views, handler and graph drawer. When the user selects the period for which prediction should be done, the handler calls the methods for relevant data to be read from the database using the **SalesDA** and passes them to the **SalesPredictor** to calculate predictions and then displays the predictions graphically. The **Promotion** package consists of the classes used to promote the items to the users and includes the view class, which identifies which month predictions should be promoted for and the number of items. The handler reads required data from the database using the **SalesDA** class and passes it to the **PromotionDA** class to publish them on the app.

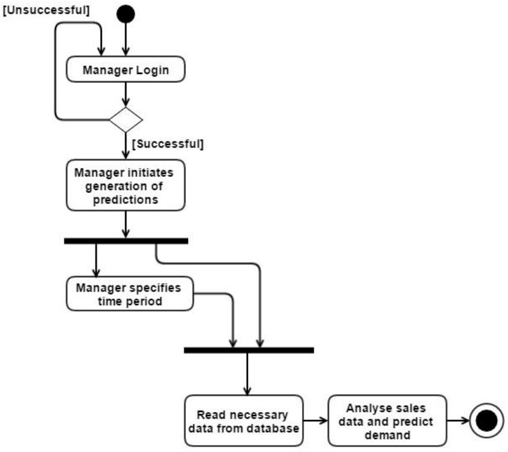

[Unsuccessful]
Manager Login
[Successful]
Manager initiates generation of predictions
Manager specifies time period
Read necessary data from database
Analyse sales data and predict demand

Fig. 3. Generate Predictions Activity Diagram.

As shown in Fig. 3, Predictions are generated by a manager by specifying the time period for which predictions need to be generated. The system would calculate and display the predictions to the user graphically.

C. Database Design

The Database Schema was developed and normalized to ensure minimum redundancy. The brand and item details were stored by storing the **brand or item ID** and the **name** of the brand/item in separate tables. A specific item of clothing of a particular brand was given a **brandItemID** and stored in a table along with the **brandID**, **itemID** and the **price** and this **brandItemID** was used for storing data relevant to the single item in a year. Sales data for a specific year are stored in a table named after the year and includes the **brandItemID**, **monthIndex** and the **sales** value for a number of items. Whenever predictions are generated they are stored in a new table created after the name of the year for which predictions are generated. **User** details including basics such as **name**, personal **details**, **username** and encrypted **password** are stored in a separate table in the database.

IV. System Implementation

A. Implementation Procedure

Sala was developed following the Rational Unified Process [5] with four basic iterations namely developing the Analysis Subsystem, developing the Prediction Subsystem, developing the Promotion Subsystem and Integration.

The standalone system was developed in Java using the NetBeans IDE [6] and JUnit [7] was used for testing. The system was developed in a front end first approach where first the user interfaces were designed and implemented after which the necessary business logic and data access classes were developed. Design Patterns such as Singleton were used with classes such as the database connection class in order to ensure best practices were followed. The system was packaged as seven main packages namely **User, Util, Sales, Analysis, Promotion**, **Prediction** and **Test**. The **User** package dealt with activities relevant to users, specifically managing user accounts. The **Util** package deals with managing the database connection and security related activities including logging in while the **Sales** package deals with the various Brands, Items, Catalogues and also managing sales data access. The **Analysis** package deals with handling the user activities in the analysis process and retrieving and presenting relevant data graphically to the user. The **Prediction** package includes the classes used to get user input and predict data using past sales data and presenting these predictions to the user graphically. Prediction was done using least square regression considering past data for several years and predicting future sales by fitting the best fit line. JFreeChart [8], an open source software distributed under GNU Lesser General Public Licence, was used to present the data graphically in both analysis and prediction. Publishing the most popular items to be displayed on the mobile app was done by the classes in the **Promotion** package. The most popular items were encoded as JSON objects and passed to and parsed in PHP to be inserted in a database to be accessed by the Android app developed separately to promote items. JUnit was used as the unit testing framework to carry out unit testing for the classes and methods of the system and UISpec4j was used for interfaces.

The mobile app was developed in Android aiming to attract a large section of the market as Android has the largest market share among mobile operating systems. Android Studio was used in the development and the networking library Volley [9] was used to make network calls to load images efficiently. The most popular items published by the promotion subsystem were read from the database and encoded in JSON format which was then retrieved using PHP to be displayed on the app. The interface was designed to be simple so that users would be able to identify just the most popular items instantly. Testing for the app was carried out with the testing facility provided with Android Studio.

Although attempts were made to use real sales data for testing purposes during the development of the system it was not possible. Therefore data had to be generated artificially. Sales data relevant to each year was stored in separate tables in the format of **brandItemID**, **monthIndex** and **sales** value. The sales values were stored as integers.

The most important algorithm used in the system could be identified as the algorithm for prediction of sales for an entire year. Prediction was done by month for the year by calculating the best fit line of least square regression in such a manner that the square of the error is minimized.

The sales values for each item for every month in every year data is available for, is stored in an instance of the **brandItem** class. The algorithm for prediction accesses these sales details stored to calculate the slope and intercept of the best fit line for a particular item. Time (year), the independent variable is coded as 1 for first year, 2 for second year, etc. to make the interpretation easier. The algorithm calculates and stores Σt, ΣY, ΣtY, Σt^2 values where t refers to the year in encoded format and Y is the sales value relevant to the year. Then it calculates the slope (1) and the intercept (2) for the line of best fit as follows:

$$b = \frac{(n\Sigma tY - \Sigma Y \Sigma t)}{(n\Sigma t^2 - (\Sigma t)^2)} \tag{1}$$

$$a = \frac{\Sigma Y}{n} - b\left(\frac{\Sigma t}{n}\right) \tag{2}$$

These values are applied to: Y' = a + bt, to identify prediction Y' where t is the encoding of the year for which predictions are done.

B. Main Interfaces

1. Desktop Application

According to the security level of the user using the system he would be able to click one or more of the buttons on the main interface to either analyse, predict, promote sales or register a user. If the manager using the system is a Sales Manager only the analysis button would be enabled, if he is a General Manager only analysis and prediction buttons would be enabled and if he is a Marketing Manager only analysis and promotion buttons would be enabled. For an admin, all buttons including the Register New User button would be enabled.

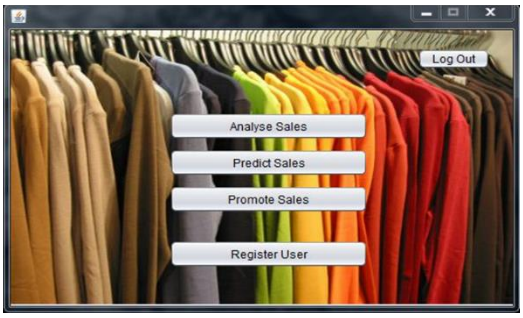
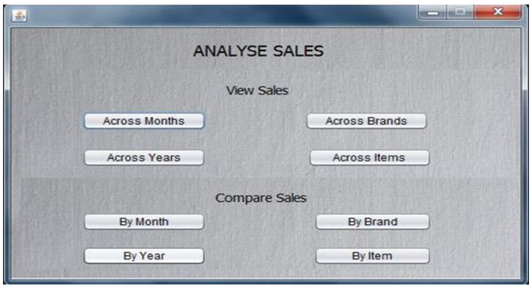


Fig. 4. Main Interface.

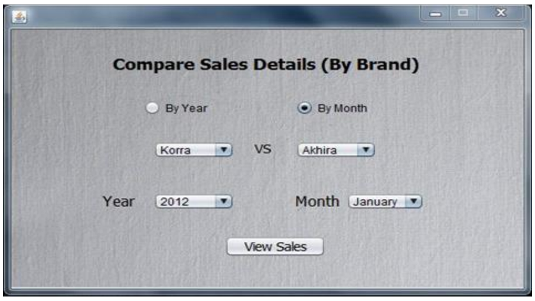

ANALYSE SALES

View Sales

Compare Sales

Fig. 5. Analyse Sales Main Interface.

If the user is authorized to analyse sales he can view sales across brands/items/months and years and compare sales. Upon clicking on a button a window (such as Fig. 6) would be displayed for the user to set necessary parameters after which required data would be retrieved and displayed graphically.


Compare Sales Details (By Brand)

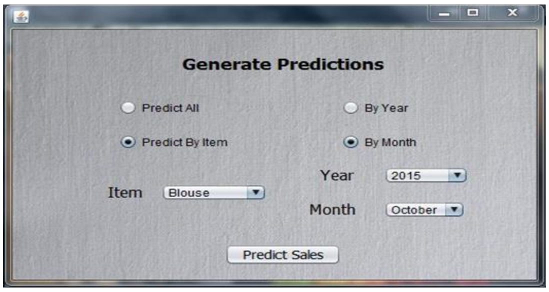
○ By Year ● By Month

Korra VS Akhira

Year 2012 Month January

Fig. 6. Compare Brand Sales Interface.

Upon clicking on the Predict Sales Button on the main interface the user would be prompted by the prediction main interface to set necessary requirements as to whether sales have to be predicted for all the items or a single item and the time period. Predictions would be calculated accordingly and displayed graphically as shown (Fig. 8).


Generate Predictions

○ Predict All ○ By Year

● Predict By Item ● By Month

Year 2015

Item Blouse

Month October

Fig. 7. Predict Sales Interface.

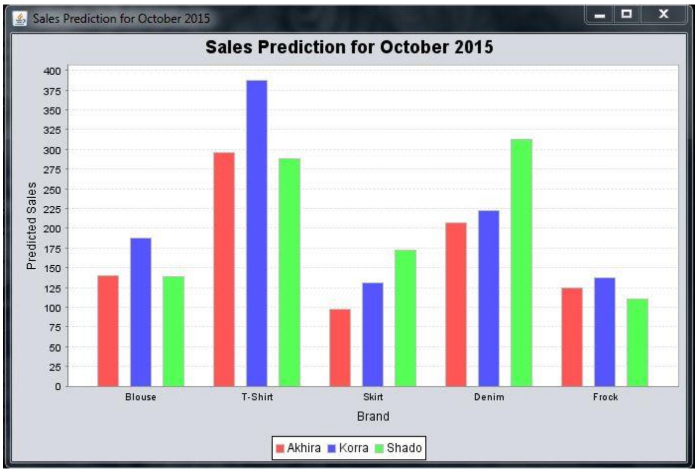
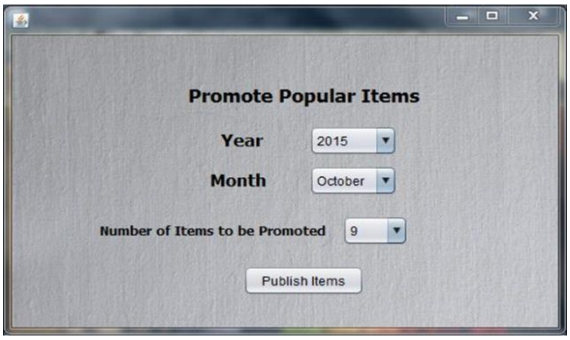


Fig. 8. Generated Predictions Interface.

Via the Promote Items interface (Fig. 9) the user would select the month for which predictions should be done and the number of items, which are most popular in that month, to be displayed via the android app. Upon publishing the said number of items are displayed on the app.


Promote Popular Items

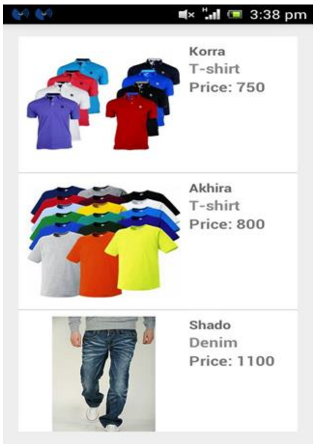
Year 2015

Month October

Number of Items to be Promoted 9

Fig. 9. Promote Items Interface.

2. Android Application



Korra
T-shirt
Price: 750

Akhira
T-shirt
Price: 800

Shado
Denim
Price: 1100

Fig. 10. Android App Main Interface.

The Android app at any given time displays the items predicted to be popular as published via the standalone application. Upon clicking on an image it would be displayed enlarged to the user.

V. System Testing and Analysis

The approach to testing was to test all components from the user interfaces to the business logic and the data access done by the system. In addition security and performance aspects of the system were also tested to ensure proper functioning.

User Interface testing was carried out to ensure that proper transitioning between windows and proper access (enabling) of UI components was allowed as expected. Business Logic classes such as the SalesPredictor class and various handlers were tested to ensure that they worked as expected. The data access done by the system was also tested to ensure the data was retrieved and inserted accurately as any shortcomings could lead to incorrect decision making ultimately leading to losses to the client.

Unit testing was the main technique used to test the classes/methods of the system. JUnit unit testing framework was used to carry out tests and test results were analysed to identify errors/shortcomings. UISpec4J was used for interface testing and JMeter for performance. Testing of the Android app was done using the testing provided with Android Studio and JUnit.

The basic getter/setter methods and the methods of the handlers were tested with unit testing to ensure proper functioning. Expected values from data access methods were calculated manually and compared with the value returned by the method and test results were analysed to identify any errors in the queries/methods and necessary adjustments were made.

Testing was carried out with regard to the SalesPredictor class's prediction methods which are of crucial importance to the system as inaccurate predictions could lead to losses to the client. Unit testing was carried out by using the prediction method to predict sales for a past month/year and comparing it against the actual sales value. Analysis of the results to identify if the predicted value fell within an acceptable range around the actual value, identified values as being acceptable confirming the acceptability of the prediction algorithm.

Security is of utmost importance to the system as any leak in sales data or publication of incorrect predictions for promotion could cause losses in sales to the client ultimately causing massive financial losses. Therefore testing to check if proper security is in place was given prominence. Users are assigned security levels according to which the functionality they can carry out is limited. Testing was carried out to ensure that UI buttons were enabled/ disabled according to the permission level of the users and users were allowed to carry out only the tasks they were authorized to. Testing was also carried out to ensure the functionality a particular user was able to carry out changed to reflect a change in his/her permission level.

VI. Conclusion and Future Work

This paper has presented a tool called Sala for sales data analytics, which is a popular field under decision support systems, where users can analyze their business domain data. This tool enables a complete solution facilitating both retail owners and customers. The customers can use this tool to identify opportunities and ultimately maximize sales and profits with more effective reach to the customers of the outlet.

The tool Sala helps to identify variations in buying patterns of customers caused due to various reasons. The analysis and prediction subsystems help the management identify how much of each item or brand is sold and what is expected to be in demand in the future. The management can thus order in such a manner that they have more of items that are predicted to sell more and less of the items that are not so popular and thereby maximize sales and profits to the company. Furthermore the promotion subsystem including the Android app serves as an ideal tool for the management to approach customers with an uncluttered user friendly app which could even increase demand for items as users can easily identify popular items.

Future work related to the system could be identified across several different avenues. The prediction mechanism could be improved to provide more consideration to the trend of sales across a year and current trends in the fashion industry when predciting demand for a month. Currently the system only considers the sales value for the particular month across several years. Another improvement could be to provide a discount code to users who use the mobile app which they could put forward to claim a discount on the items advertised. This would encourage more customers to use the app and thereby increase the demand, sales and profits even more. In general as an overall improvement the user interfaces could also be improved to provide a better user experience. Overall it could be concluded that Sala has the potential to improve and maximize the sales and profitability of the client.

References

[1] Small Business – Chron.com, "Examples of Seasonal Inventory", 2015. [Online]. Available: http://smallbusiness.chron.com/examples-seasonal-inventory-61418.html

[2] Thomassey S., "Sales Forecasting in Apparel and Fashion Industry: A Review", 2014

[3] Liu N. et al., "Sales Forecasting for Fashion Retailing Service Industry: A Review", *Mathematical Problems in Engineering*, 2013

[4] Huang W. et al., "A Novel Trigger Model for Sales Prediction with Data Mining Techniques", *Data Science Journal* 14 (2015)

[5] "Rational Unified Process: Best Practices for Software Development Teams", 2011. [Online]. Available: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

[6] "NetBeans IDE Features". [Online]. Available: https://netbeans.org/features/

[7] "JUnit: About". [Online]. Available: http://junit.org/

[8] "JFreeChart". [Online]. Available: http://www.jfree.org/jfreechart/

[9] "Transmitting Network Data Using Volley". [Online]. Available: http://developer.android.com/training/volley/index.html