Design & Implementation of a Web Application for Microfinance Institution Financial Ratio Reporting

Opara Febechukwu Chinonyerem
elahsoft@gmail.com,oparafebechukwu@gmail.com

Abstract— Microfinance Financial Ratio reporting is very essential as they help the management in seeing into the future of their institution and determining the current state of their institution. This research paper developed a web application that assists the personnel responsible for reporting these ratios to prepare the reports and also keep a record of the reports for easy referencing during comparison of past periods and the present period. The Model View Controller (MVC) approach was used to develop the project, Agile (SCRUM) was used as the project management style, and Test Last Development was used as the development approach. The third party libraries used in developing and testing the system include Codeception (for tests), Bootstrap (for styling), FontAwesome for icons, DataTables for tables and FPDF (for generating the report in a pdf format), The system can be upgraded to become an expert system which studies the trend of the financial ratios and alerts the management on failures and good points in the financial state of their institution.

Index Terms— Financial Ratios, Ratio Analysis, Sustainability, Profitability, Efficiency, Productivity, Agile, MVC, TLD, SCRUM

1 INTRODUCTION

The Management of a Microfinance Institution is a very tough one. Financial statements which are products of Financial Accounting are efficient in revealing information about the state of affairs of the institution but they do not equip the management efficiently for apt financial decision making; hence the need for Management Accounting. There are various Management Accounting Components, they include: Ratio Analysis, Cost Volume Price Analysis (CVP Analysis), Costing, Transfer Pricing, Capital Budgeting, and Sales and Financial Forecasts.

Financial Ratios are simply figures gotten by dividing a numerator by a denominator. They are employed in the management of Microfinance Institutions to determine the profitability, sustainability, efficiency and productivity of the Microfinance Institution. The Management of the Microfinance Institution analyses this ratio, that is studies the trend of the past period's ratio and the present one, and determines if the Institution is sustainable, efficient, profitable and productive, this process is what is referred to as Ratio Analysis.

Currently, most Microfinance Institutions in Nigeria prepare this financial ratios report manually, which is tenable to errors and most leave out some of the ratios considering them as not necessary. The financial ratios must be analyzed together, as selective ratio use creates an incomplete picture of the financial health of the institution [1]. A financial ratio should never be taken in isolation from other ratios – together, they provide a more complete picture [4]. Also, overtime as these reports grow in numbers it may become difficult to compare past period ratios to present period ratios to determine the trend for effective management decision making, hence the need for an automated system to document and prepare this financial ratios report.

2 LITERATURE REVIEW

Financial Analysis is defined by Consultative Group to Assist the Poorest (CGAP) as the computation of analytical ratios from financial statements and the interpretation of these ratios to determine their trends as a basis for management decisions [2]. The products of Financial Analysis are Financial Ratios. Financial ratios enable the manager to evaluate the performance of their organization in several different aspects of its activity [1]. They also enable a Microfinance Institution to benchmark itself with other institutions e.g. by reporting to the Microfinance Information Exchange (MIX) a microfinance bank can determine if its recovery rate or its portfolio at risk is close to the industry average for its size, its maturity, its country, or its region etc. [2]. Financial Ratio Analysis helps to answer two primary questions that every institution involved in microfinance needs to ask: Is this institution either achieving or progressing towards profitability? How efficient is it achieving its objectives? [2] Managers are advised to calculate ratios/indicators monthly or quarterly [1]. Each ratio is one important aspect of the financial organization but when the entire ratios are combined the key areas give a well-rounded global perspective of the financial situation [4]. The Small Enterprise Education and Promotion Network (SEEP) 18 financial ratios or indicators are grouped into four (4) major groups which include:

1. Profitability and Sustainability
2. Asset/Liability Management
3. Portfolio Quality
4. Efficiency and Productivity [1]

Financial Ratios or Indicators can also be grouped into five (5) major groups, which include:

1. Profitability and Sustainability
2. Efficiency and Productivity
3. Portfolio Quality
4. Growth and Outreach
5. Financial Structure [3].

2.1 Profitability and Sustainability

Profitability and Sustainability Ratios are the most comprehensive ratios [3]. They reflect the Microfinance Bank's ability to continue operating and growing in the future [2]. It shows the financial returns to the use of the capital and assets employed. It answers the question: Will our Microfinance Institution have the financial resources to continue serving members and clients tomorrow as well as today? [4] There are five basic profitability and sustainability ratios, which include:

1. (Adjusted) return on assets (ROA)
2. (Adjusted) return on equity (ROE)
3. Operational Sustainability
4. Financial Sustainability
5. Profit Margin [3].

2.1.1 (Adjusted) Return on Assets (AROA)

Return on Assets measures net operating income as a percentage of average total assets. It reveals how well the MFI uses its assets to generate a profit [3]. A positive correlation exists between this ratio and Portfolio to Assets; the ratio is higher for institutions that maintain a large percentage of the assets in the Gross Loan Portfolio [2]. Institutions that maintain most of their assets in the loan portfolio tend to break even sooner, and generate higher returns on their assets; provided the loan portfolio performs well and other costs are also controlled [4]. The Break Even Point of a Microfinance Institution is reached when enough loans with specific terms generate income that exactly equals expenses; in other words, net profit is zero, and the entity is 100 percent sustainable.

Adjusted Return on Assets (AROA) shows how efficiently the assets are used to generate profits relative to inflation [2]. An increasing Return on Assets (ROA) is positive [4]. ROA/AROA is net of taxes and excludes non-operating items and donations [1]. The equation for Return on Assets and Adjusted Return on Assets are given as:

Return on Assets = $\frac{Net\ Operating\ Income - Taxes}{Average\ Assets}$

Adjusted Return on Assets = $\frac{Adjusted\ Net\ Operating\ Income - Taxes}{Adjusted\ Average\ Assets}$

2.1.2 (Adjusted) Return on Equity (AROE)

In a for-profit Microfinance Institution as the Nigerian Microfinance Banks, Return on Equity (ROE) is the most important profitability indicator; it measures the Microfinance Bank's ability to reward its shareholders' investment, build its equity base through retained earnings and raise additional equity investment [2]. For a non-profit Microfinance Institution, Return on Equity (ROE) indicates its ability to build equity through retained earnings, and increased equity enables the Microfinance Institution to leverage more financing to grow its portfolio [2]. By excluding donations and non-operating revenues, this ratio demonstrates an institution's ability to generate income from its core financial service activity [2]. Return on Equity (ROE) is a good indicator of how well the Microfinance Institution has used retained earnings and donor money to become sustainable. As markets mature and competition increases, Return on Equity may level off and maintain a positive position without increasing dramatically or at all [4]. An increasing Return on Equity is positive [4]. Adjusted Return on Equity (AROE) is a good proxy for commercial viability, and most investors prefer it [2]. AROE is frequently used as a proxy for commercial viability because the numerator does not include non-operating items or donations and is net of taxes [1]. AROE is useful for comparing one Microfinance Institution to another because it eliminates the positive effects of subsidies and preferential borrowing arrangements and applies a minimum standard for treating a portfolio at risk. The equation for Return on Equity and Adjusted Return on Equity is given below:

Return on Equity (ROE) = $\frac{Net\ Operating\ Income - Taxes}{Average\ Equity}$

Adjusted Return on Equity (AROE) = $\frac{Adjusted\ Net\ Operating\ Income - Taxes}{Adjusted\ Average\ Equity}$

An Average is calculated by adding the beginning and ending figures of the period and dividing by two [3].

2.1.3 Operational Sustainability

It is also known as Operational Self Sufficiency (OSS). It is simply operating revenue as a percentage of operating and financial expenses. If the ratio is greater than 100 percent, it means that the Microfinance Institution is covering all of its costs through its own operations and is not relying on contributions to survive [3]. A ratio of 100% is the Microfinance Institution's break-even point and it indicates that the Microfinance Institution's income is equal to its operating expenses [4]. OSS does not tend to fluctuate as much as other ratios and the positive trend can be achieved through growth and increased efficiency. As a result, OSS is the one profitability measurement that is worth monitoring on a monthly basis. Managers should consider the drivers behind OSS – is it due to larger loan sizes, high yields, low financial expenses or efficient operations? Although not considered as rigorous indicator as AROE, OSS is a simple and useful measurement for Microfinance Institutions managers, particularly for young Microfinance Institutions that want to monitor their path to sustainability [2]. The equation for Operational Self Sufficiency is given below:

$$\text{Operational Self Sufficiency} = \frac{\mathit{Financial\ Revenue}}{\mathit{Financial\ Expense} + \mathit{Loan\ Losses} + \mathit{Operating\ Expenses}}$$

Financial Revenue is simply revenue from interest earned, fees and commissions (including late fees and penalties) on gross loan portfolio only. Interest accrued but not yet paid should also be included in the Financial Revenue [2]. Financial Expense includes all interest, fee and commissions incurred on all liabilities including deposit accounts of clients held by the institution, commercial and concessional borrowings, mortgages and other liabilities. It also includes accrued interest as well as cash payment of interest [2]. Loan Losses Provision Expense (LLPE) is a non-cash expense that is charged on the income statement and is calculated as a percentage of the value of gross loan portfolio that is at risk of default [2]. Operating Expenses include personnel expenses and administrative expenses but excludes financial expenses and loan-loss expense [2].

2.1.4 Financial Sustainability

It is also known as Financial Self Sufficiency (FSS). It is inflation and subsidy adjusted Operational Sustainability [3]. FSS is a better indicator compared to Operational Self Sufficiency to determine whether or not the income of the institution is able to cover the costs – both direct and indirect [2]. When an institution is consistently reaching FSS greater than 100 percent, managers should pay attention to AROE and AROA, which are more commercial measurements of performance [2]. Broadly speaking, OSS measures ability to survive, while FSS is a better indicator of ability to grow [1]. The equation for Financial Self Sufficiency is given below:

$$\text{Financial Self Sufficiency} = \frac{\mathit{Financial\ Revenue}}{(\mathit{Adjusted\ Financial\ Expense} + \mathit{Adjusted\ Net\ Impairment\ Losses\ on\ Loans} + \mathit{Adjusted\ Operating\ Expense})}$$

2.2 Efficiency and Productivity

Efficiency and Productivity indicators reflect how efficiently a Microfinance Institution is using its resources, particularly its assets and personnel [2]. Microfinance Institutions use different efficiency and productivity indicators, tailoring them to reflect their own organizational structure, product lines and monitoring priorities [2]. In a competitive environment, efficiency and productivity are the most important means to gain a competitive edge. Decisions about credit methodology, credit terms (loan size, maturities and so on), and markets in which to operate also affect efficiency and productivity [3]. Efficiency ratios should be examined in conjunction with Portfolio Quality ratios, since a quality portfolio should not be sacrificed for high efficiencies [4]. The Efficiency and Productivity ratios include:

1. (Adjusted) Operating Expense Ratio
2. (Adjusted) Cost per Active Client
3. Borrowers per Loan Officer
4. Active Clients per Loan Officer
5. Client Turnover
6. (Adjusted) Average Outstanding Loan Size
7. Average Loan Disbursed [1].

The Nigerian Microfinance Certification Programme (MCP) identified two Efficiency and Productivity ratios, they include:

1. Operating Expense Ratio
2. Cost per Client [2].

Mennonite Economic Development Associates identified the following Efficiency and Productivity ratios:

1. Operating Expense Ratio
2. Cost per Active Client
3. Borrowers per Loan Officer
4. Active Clients per Staff Member
5. Client Turnover
6. Average Outstanding Loan Size
7. Average Loan Disbursed [4].

2.2.1 (Adjusted) Operating Expense Ratio

This highlights personnel and administrative expenses relative to the loan portfolio the most commonly used efficiency indicator [1]. The Operating Expense Ratio enables managers to quickly compare administrative and personnel expenses to the Microfinance Institution's yield on gross loan portfolio [2]. For this reason, the Operating Expense ratio is frequently referred to as the *efficiency ratio* [2]. Monitoring this trend is an easy way to observe if the Microfinance Institution is increasing its efficiency as it grows its loan portfolio [2]. The lower the ratio, the more efficient the Microfinance Institution is [2]. Microfinance Institution should strive for a downward trend in this ratio – even when portfolio growth is flat – until they are convinced that no more efficiency can be found [2]. This ratio may fluctuate from month to month, but it should decline from year to year [2]. This ratio goes hand in hand with the portfolio yield (financial income/average portfolio) [3]. For instance, if the portfolio yield is 50 percent, then the Microfinance Institution is striving to have an operating expense ratio lower than 50 percent [3]. An increase in average loan size will also create a decreasing trend, but not necessarily signal increased efficiencies [4]. A decreasing operating expense ratio is positive [4]. The adjusted ratio increases this ratio when the effects of subsidies are included [1].

Managers should pay close attention to the adjusted ratio because it reveals how efficient the MFI truly is when all subsidies are removed [1]. The equation for operating expense ratio and adjusted operating expense ratio is given below:

Operating Expense Ratio = $\frac{Operating\ Expense}{Average\ Gross\ Loan\ Portfolio}$

Adjusted Operating Expense Ratio = $\frac{Adjusted\ Operating\ Expense}{Average\ Gross\ Loan\ Portfolio}$

2.2.2 (Adjusted) Cost per Active Client

This ratio provides a meaningful measure of efficiency for a Microfinance Institution; it allows the Microfinance Institution to determine the average cost of maintaining an active client [1]. In concrete terms, Cost per Client Ratio indicates to an institution how much it currently spends in Personnel and Administrative Expenses to serve a single active client. It informs the Microfinance Institution how much it must earn on average from each client to be profitable [2]. It is the amount of revenue that the Microfinance Institution needs to generate from every single client in order to break-even [4]. By examining the trends in the ratio's numerator and denominator, and also the ratio itself, one can determine whether a reduced Cost per Client is the result of reduced labor costs or the more efficient use of labor [4]. The equation for Cost per Active Client and Adjusted Cost per Active Client are given as:

Cost per Active Client = $\frac{Operating\ Expense}{Average\ Number\ of\ Active\ Clients}$

Adjusted Cost per Active Client = $\frac{Adjusted\ Operating\ Expense}{Average\ Number\ of\ Active\ Clients}$

2.2.3 Borrowers per Loan Officer

The ratio measures the average number of borrowers managed by each loan officer [1]. This ratio reflects the productivity of loan staff in serving their client caseload. The higher the caseload per officer, the more clients will be served, and the greater the efficiency gained. The ratio is useful when the MFI sets growth targets; it is easy to calculate and to measure on a regular basis [4]. The equation is given as:

Borrowers per Loan Officer = $\frac{Number\ of\ Active\ Borrowers}{Number\ of\ Loan\ Officers}$

2.2.4 Active Clients per Staff Member

This ratio shows the overall productivity of the Microfinance Institution's personnel in terms of managing clients, including borrowers, voluntary savers, and other clients [1]. An increasing trend is desirable. The ratio does not differentiate between credit or service staff and administrative staff. A low ratio however does not necessarily mean that staff is not working hard. There are many factors affecting both service and administrative productivity, not to mention paperwork, technology, remote service locations, etc. What is important to remember is to regularly monitor this ratio, particularly in relation to the MFI's business plan [4]. The equation for this ratio is given as:

Active Clients per Staff Member = $\frac{Number\ of\ Active\ Clients}{Numbe\ rof\ Personnel}$

2.2.5 Client Turnover

The ratio is used as one measure of client satisfaction [1]. This ratio is intended to give Microfinance Institution's managers some indication of clients who leave the program during a particular period of time. It does not tell management why clients may leave. The ratio is not precise or perfect – it does not provide information about clients who leave for a while and then return at a later date. It does not provide information about clients who may use a remittance service occasionally after no longer using lending services. The ratio is considered important because managers need some level of client satisfaction with the MFI's products and services. The general thinking is that cost of retaining clients is relatively lower than the cost of recruiting and initiating new clients to the MFI. Therefore the lower the turnover, the higher perceived level of satisfaction of clients [4]. The equation is given as:

Client Turnover = $\frac{\begin{matrix}Number\ of\ Active\ Clients\ at\ beginning\ of\ period+\\ Number\ of\ New\ Clients\ during\ period+\\ Number\ of\ Active\ Clients\ at\ end\ of\ period\end{matrix}}{Average\ Number\ of\ Active\ Clients}$

2.2.6 (Adjusted) Average Outstanding Loan Size

This ratio is a profitability driver and a measure of how much of each loan is available to clients [1]. For many, this ratio is some indication of the MFI's ability to reach as many lower-income and poor clients as possible [4]. From a financial perspective, this ratio can be used to project and plan portfolio growth as loan size is one of the drivers of profitability [4]. This ratio may fluctuate for many reasons – seasonality, change in methodology, disbursements patterns and loan size. It is important to compare against the MFI's business plan. The adjusted ratio is the Average Outstanding Loan Size adjusted by the effect of Write-offs [1]. The equation is given as:

Average Outstanding Loan Size = $\frac{Gross\ Loan\ Portfolio}{Number\ of\ Loans\ Outstanding}$

Adjusted Average Outstanding Loan Size = $\frac{Adjusted\ Gross\ Loan\ Portfolio}{Number\ of\ Loans\ Outstanding}$

2.2.7 Average Loan Disbursed

This ratio measures the average value of each loan disbursed. This ratio is frequently used to project disbursements [1]. This ratio gives MFI managers some idea about the MFI's demand for loans, client profitability and the capacity to increase income. This ratio should be monitored in conjunction with portfolio quality ratios, to ensure that the client still has the capacity to repay the loan, in spite of increased amounts of debt [4]. This ratio may fluctuate for many reasons – seasonality, change in methodology, disbursements patterns, increasing capacity of clients to handle debt, the growth of client businesses, high client retention, and automatic loan size increases, to name a few [4]. The equation is given as:

Average Loan Disbursed = $\frac{Value\ of\ Loans\ Disbursed}{Number\ of\ Loans\ Disbursed}$

2.3 Portfolio Quality Ratios

The primary asset of a Microfinance Institution is its gross loan portfolio. Portfolio quality is important to the financial success of any microfinance institution. Drops in portfolio quality could mean a decline in customer satisfaction and, therefore, may presage a low retention rate resulting in higher costs to recruit new clients. Drops may also signal problems in staff supervision and control [1]. This group of ratios measures the “health” of the loans outstanding in terms of its risk. It answers the question: Is our MFI responsible in managing our most important asset? [4]. It is crucial that board members know the details of the quality of the portfolio. Therefore, a portfolio report should always be included with company reports. Management, too, needs this data to run the institution [3]. The three Portfolio Quality Ratios include:

1. Portfolio at Risk
2. Write-off Ratio
3. Risk Coverage Ratio

2.3.1 Portfolio at Risk (PAR)

Portfolio at Risk (PAR) is important because it indicates the potential for future losses based on the current performance of the loan portfolio. The PAR ratio is the most widely accepted measure of loan performance in the microfinance industry. PAR > 30 days is often used as the threshold beyond which loans are considered to be at higher risk. This ratio also includes Renegotiated Loans. This not only prevents hiding troubled loans through rescheduling or refinancing, but also indicates a higher level of risk associated with clients who have had repayment problems [1]. Best practice and regulatory threshold for Nigeria requires that PAR for microfinance banks should not exceed 2.5% [2]. A decreasing Portfolio at Risk is positive [4]. The ratio takes the Portfolio at Risk over 30 days late and adds the Renegotiated Loans in the numerator. Loans delinquent for up to 30 days generally have high potential of being collected, with the appropriate management approach. However, Renegotiated Loans are at already at risk, since they have already had repayment problems [4]. The equation is given as:

Portfolio at Risk = $\frac{Portfolio\ at\ Risk>30\ days+Renegotiated\ Loans}{Gross\ Loan\ Portfolio}$

Or

Portfolio at Risk = $\frac{Sum\ of\ Principal\ outstanding\ on\ all\ past-due\ loans+Renegotiated\ Loans}{Gross\ Loan\ Portfolio}$

2.3.2 Write-off Ratio

Write-offs are an accounting device to remove persistently delinquent loans from the books; this ratio is highly dependent on a Microfinance Institution’s write-off policy. Managers should monitor this ratio as frequently as write-offs are made. A Microfinance Institution should disclose its write-off policy, which should be clearly defined, followed, and monitored by the board. A Microfinance Institution’s write-offs should be minimal, however; most MFI’s write-off ratios are remarkably low [1]. Write-offs pose the greatest risk to an MFI because they are loans that a Microfinance Institution has determined are uncollectable. Not only will the Microfinance Institution forgo financial income on a written-off loan, it may also lose the loan principal that is outstanding. Large write-offs can reflect poor management procedures for client selection and loan approvals, or for the collection of delinquent loans [3]. An aggressive write-off policy reduces the PAR Ratio. For this reason, the PAR Ratio and Write-off Ratio should be viewed together. An aggressive write-off policy should be accompanied by strong loan recovery efforts [2]. The equation is given below:

Write-off Ratio = $\frac{Value\ of\ Loans\ written\ off}{Average\ Gross\ Loan\ Portfolio}$

2.3.3 Risk Coverage Ratio

The Risk Coverage Ratio measures how adequate the Impairment Loss Allowance is to account for potential loan losses. Because the Impairment Loss Allowance represents the Microfinance Institution’s preparation for loan losses, the Risk Coverage Ratio is an approximate indicator of how prepared an institution is to absorb loan losses in the worst-case scenario; that is, if all Portfolio at Risk > 30 days became uncollectible. Although the Impairment Loss Allowance is expressed as a negative number on the balance sheet, the Risk Coverage Ratio is expressed as a positive percentage [1]. A fairly constant, stable ratio is desired. Sudden changes usually indicate a deterioration or improvement in portfolio quality or an excess or shortage in the Allowance for Loan Losses account [4]. The equation is given as:

Risk Coverage Ratio = $\frac{Allowance\ for\ Loan\ Losses}{Portfolio\ at\ Risk\ over\ 30\ days}$

2.4 Asset and Liability Management Ratios

These ratios help to manage the use of funds and source of funds efficiently, the goal being to use the funds borrowed to generate funds that exceed the cost of the funds. Asset/liability management is required on the following several levels:

- **Interest Rate Management.** The Microfinance Institution must make sure that the use of funds generates more revenue than the cost of funds.

- **Asset Management.** Funds should be used to create assets that produce the most revenue (are most "productive").
- **Leverage.** The Microfinance Institution seeks to borrow funds to increase assets and thereby increase revenue and net profit. At the same time, the Microfinance Institution must manage the cost and use of its borrowings so that it generates more revenue than it pays in Interest and Fee Expense on those borrowings. It must also be wary of borrowing more than it can repay in times of trouble.
- **Liquidity Management.** The Microfinance Institution must also make sure that it has sufficient funds available ("liquid") to meet any short-term obligations.
- **Foreign Currency Management.** This occurs when a Microfinance Institution lends, invests, or borrows in a foreign currency. The risk is greatest when a mismatch exists between the value of foreign currency assets and liabilities, leaving the Microfinance Institution exposed to either a decline in the value of its assets or an increase in balance of its funding liabilities when a significant movement in exchange rates occurs [1].

The Asset and Liability Management Ratios include:

1. Yield on gross Portfolio
2. Portfolio to Asset
3. Cost of Funds Ratio
4. Debt to Equity Ratio
5. Liquidity Ratio

2.4.1 Yield on Gross Portfolio

Yield on Gross Portfolio, also called portfolio yield, measures how much the Microfinance Institution actually received in cash interest payments and Fees and Commissions from its clients during the period; this ratio is the initial indicator of an Microfinance Institution's ability to generate cash for operations from the Gross Loan Portfolio. Cash receipts from the Gross Loan Portfolio are vital for a Microfinance Institution's survival [1]. Generally, the Microfinance Institution's yield does not fluctuate very much, unless there is a change in interest rates, the method of calculating interest or in fees. If it varies significantly from the effective interest rate (often called a "yield gap") then it may be a sign of delinquency. However, the portfolio yield cannot exceed the effective rate of interest charged by the institution [4]. An increasing yield is positive although it will level off as it nears the effective interest rate [4]. The equation is given as:

Yield on Gross Portfolio = $\frac{Cash\ received\ from\ Interest, Fees, and\ Commissions\ on\ Loan\ Portfolio}{Average\ Gross\ Loan\ Portfolio}$

2.4.2 Portfolio to Asset

A Microfinance Institution's primary business is granting loans and providing other financial services to micro-entrepreneurs. This ratio shows how well a Microfinance Institution allocates its assets to its primary business and, in most cases, its most profitable activity—granting loans [2].

At first glance, a manager can see how well the Microfinance Institution is deploying its funds into high-yielding microloans. This ratio is most valuable when observed monthly. The Gross Loan Portfolio can fluctuate dramatically month to month if the Microfinance Institution experiences seasonal spikes in demand for loans. Managers can also use the ratio to identify fluctuations that may result from structural or operational rigidities that cause a high number of loans to be disbursed or repaid at the same time. Depending on the context, this ratio could indicate the need for additional funding or be a sign of excess liquidity [1]. Microfinance Institutions that rely heavily on savings to fund their portfolio tend to be more efficient at maintaining a high and steady Portfolio to Assets ratio [2]. An increasing trend is positive [4]. The equation is given as:

Portfolio to Assets = $\frac{Gross\ Loan\ Portfolio}{Assets}$

2.4.3 Cost of Funds Ratio

This ratio gives a blended interest rate for all the Microfinance Institution's average funding liabilities, deposits, and borrowings. The denominator does not include other liabilities, such as Accounts Payable or a car loan [2]. When compared to Yield on Gross Loan Portfolio, it reveals how the cost of funding the Gross Loan Portfolio with borrowings relates to the Yield on the Gross Loan Portfolio. This relationship is the key element of successful interest rate management [1]. The Cost of Funds may indicate a level of maturity of the institution. A decreasing Cost of Funds ratio is generally positive. When Financial Expenses are adjusted to include free or subsidized funding, the ratio will show the actual financial cost of funds needed to fund or capitalize the Microfinance Institution [4]. The equation is given as:

Cost of Funds = $\frac{Financial\ Expense\ on\ Funding\ Liabilities}{(Average\ Deposits+Average\ Borrowings)}$

2.4.4 Debt to Equity Ratio

The Debt to Equity ratio (also referred to as Debt/Equity ratio or the Leverage ratio) is a common measure of a Microfinance Institution's capital strength or adequacy at a particular point in time [4]. An increasing debt/equity ratio indicates the Microfinance Institution's capacity to attract debt funding based on the capital strength of its own equity. Too low a ratio might indicate that the Microfinance Institution is not maximizing its equity base. Too high a factor may be risky for investors, and may spell cash flow challenges during difficult times [4]. In Nigeria there are no regulatory restrictions on the debt/equity ratio; it is however good practice not to have more than a debt equity ratio of 1:2 [2]. The equation is given as:

Debt to Equity Ratio = $\frac{Liabilities}{Equity}$

2.4.5 Liquidity Ratio

The Liquidity Ratio is one measurement of the sufficiency of cash resources to pay the short-term obligations to depositors, lenders, and other creditors [2]. The Liquidity Ratio can help managers monitor the sufficiency of cash to meet the most immediate obligations, namely Short-term Deposits and Borrowings and other short-term payables and liabilities [2]. No single ratio or trend provides the "correct" or "adequate" means to monitor cash levels. Managers must have clear policies in place to ensure that cash is available when needed for all Microfinance Institution's operations and activities. Banking requirements and risk tolerance will affect the ratio [4].

3 SYSTEM DEVELOPMENT METHODOLOGY

The project management style adopted is Agile. Agile represents a group of software engineering methodologies which promise to deliver increased productivity, quality and project success rate overall in software development projects [5]. Agile is a project management style in which the project is broken down into small manageable chunks called iterations. At the end of each iteration something valuable is delivered, which can be presented to the project stakeholders for feedback. Some of the Agile software development methodologies are SCRUM, Extreme Programming (XP), Crystal and Kanban. The Agile methodology adopted for the project is the SCRUM. Scrum is an Agile software development methodology in which the project is broken down into short iterations of one to three weeks. Each short iteration is called a sprint. The life cycle of each sprint includes:

1. Planning
2. Execution
3. Review
4. Rinse and Repeat, [6].

A project backlog is created for every Scrum project. A project backlog is a list of all the product features generally defined by user stories [6]. User stories define everything potential users want to do on the site [6]. User stories are structured as:

AS A [type of user]

I WANT TO [do this thing]

SO THAT I CAN [accomplish this goal], [6].

4 SYSTEM ANALYSIS & DESIGN

There are only two users of the system, which is a client and the administrator of the system. The client does basically nothing than viewing pages that provide a simple guide on what each menu item does, whereas the administrator can do much more. The project backlog is given by table 1 below:

Table 1: The project backlog.

SPRINT	AS A/AN	I WANT TO	SO THAT I CAN
1	Client (Ordinary user)	Be able to view the log in page and be able to view the purpose of each menu item	So that I can prepare the report and be guided on what to use each hyperlink for.
	Administrator	To view links for Sustainability Reports	So that I can compute Financial Self Sufficiency (FSS), Operational Self Sufficiency (OSS), Adjusted Return on Assets (AROA) and Adjusted Return on Equity (AROE).
	Administrator	To view the generate report page	So that I can view the Financial Ratio report sheet reporting the OSS, FSS, AROA and AROE values and corresponding notes about the values.

2	Administrator	To view links for Asset/Liability Management ratios	So that I can compute Yield on Gross Loan Portfolio, Portfolio to Asset Ratio, Cost of Funds, Debt to Equity Ratio and Liquidity Ratio
		To view the generate report page.	So that I can view the Financial Ratio report sheet reporting Yield on Gross Loan Portfolio, Portfolio to Asset Ratio, Cost of Funds Ratio, Debt to Equity Ratio and Liquidity Ratio.
3	Administrator	To view links for Portfolio Quality and Efficiency/Productivity ratios	So that I can compute Portfolio at Risk(PAR), Write-off Ratio, Operating Expense Ratio and Cost per Client Ratio
		To view the generate report page.	So that I can view the Financial Ratio report sheet reporting Portfolio at Risk, Write-off Ratio, Operating Expense Ratio and Cost per Client Ratio.

5 SYSTEM IMPLEMENTATION

The system was implemented using PHP as the server-side programming language, Javascript as the client-side programming language, CSS as the styling language, MYSQL as the database platform and HTML as the scripting language. The third party libraries used for the project include:

1. Bootstrap CSS for styling.
2. DataTables for HTML Tables.
3. FontAwesome for Icons
4. FPDF for generating the report in PDF format.

The development style adopted is Test Last Development (TLD). Test Last Development involves writing the codes for the feature before the tests.

The data flow diagram for the proposed system is given in figure 1. The database entity relationship diagram for the proposed system is given in figure 2. The tables in the MYSQL database (named ratio_analysis) include: ra_users, ra_adjusted_return_on_assets, ra_adjusted_return_on_equity, ra_cost_of_funds, ra_cost_per_client, ra_debt_to_equity, ra_financial_self_sufficiency, ra_liquidity_ratio, ra_management_report_period, ra_operational_self_sufficiency, ra_portfolio_at_risk, ra_portfolio_to_asset, ra_write_off_ratio and ra_yield_on_gross_portfolio. The ra_users stores the administrator's username and password and the rest of the tables store the user's inputted quantities for computing the different financial ratios. The models for the application are stored in the models folder, the controllers in the controllers folder and each controller is prefixed with a c_ and the view are stored in the admin folder and are prefixed with a v_.

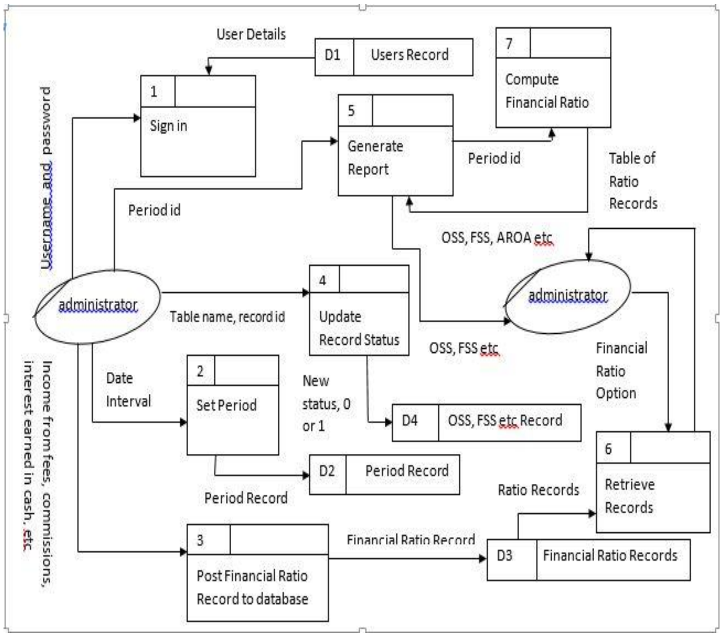

Figure 1: Data Flow Diagram of the Proposed System

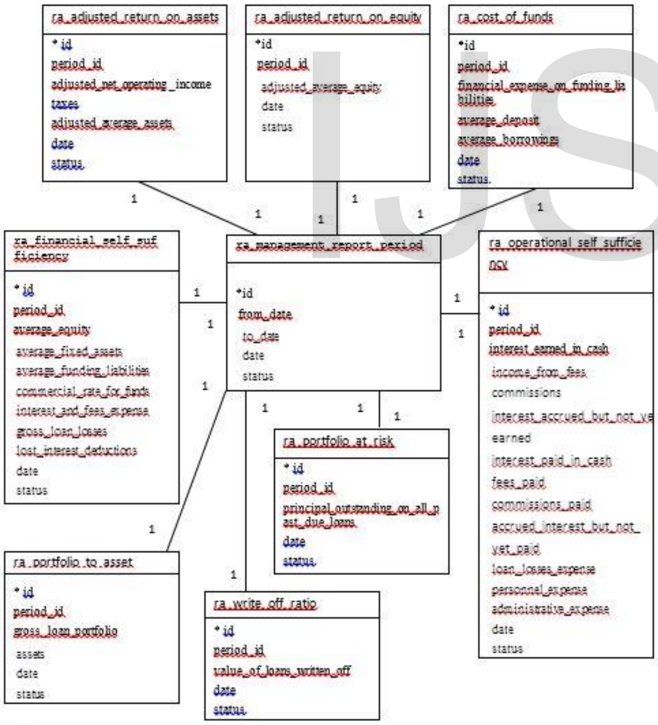

Figure 2: Entity Relationship Diagram of the proposed system.

The rest of the tables not depicted in the diagram above, have a one to one relationship with the ra_management_report_period table and the ra_management_report_period table has a one to many relationship with other tables. That is each table houses a period_id field which is the id of a record in the ra_management_report_period table. The detailed code and graphic user interface for the proposed system can be found in the following github link: https://github.com/elahsoft/DESIGN-AND-IMPLEMENTATION-OF-A-WEB-APPLICATION-FOR-MICROFINANCE-FINANCIAL-RATIO-REPORTING-SYSTEM.

6 SYSTEM TESTING

The tests covered are acceptance test and unit test. Acceptance test are tests carried out to ensure that the feature to be coded must be acceptable by the client. Unit tests are tests carried out to ensure that each unit code written is behaving the way it is expected to behave. Functional testing wasn't done because we are not using a framework like Symfony, YII2, Laravel etc. The testing framework adopted for the acceptance and unit tests is the PHP testing framework called Codeception.

6.1 Acceptance Test

Acceptance tests try to emulate the behavior of a real user using the application. The tests are ran on PhpBrowser. PhpBrowser is the fastest way to run acceptance tests. It is simply a PHP script that can check HTML page contents, click links, fill forms and submit POST and GET requests, but for more complex tests that require a browser or that involves Javascript codes, Selenium with WebDriver module is used. To install Codeception, open the command prompt, go to the root folder of the application and do the following:

1. Download composer for your platform, Windows 10 was used for the project.
2. Install composer and use it to install Codeception at the root folder of the app using the command: composer require "codeception/codeception" –dev

The steps taken to set up and run an acceptance test include:

1. In the root folder of the application, run the command: vendor\bin\codecept bootstrap. This command creates the configuration file codeception.yml, tests directory and default test suites i.e. unit, acceptance and functional test suites.
2. Generate an acceptance test file using the command: vendor\bin\codecept generate:cest acceptance Signin. Signin here is the filename for the test file to be generated by Codeception.
3. Start the local development server, and put the application's url into the acceptance.suite.yml file, located in the tests directory.
4. Write the tests and run with the command vendor\bin\codecept run –steps. The command runs all the tests showing the steps taken down the execution, to run only the acceptance tests i.e. ignoring the unit tests you use the command: vendor\bin\codecept run acceptance –steps, to run a specific file you use the command vendor\bin\codecept run acceptance SigninCest –

steps, where SigninCest is the filename of the file bearing the tests.

The tests now runs with PhpBrowser.

6.2 Unit Test

The command for generating a unit test file is: vendor\bin\codecept generate:test unit Signin. Signin is the filename of the test. The file created for the test will have a suffix Test whereas that of acceptance test have the suffix Cest. The functions that contain the unit tests have the prefix test. To run all the unit tests the command used is: vendor\bin\codecept run unit –steps.

7 CONCLUSION

This paper centered on building a web application for Microfinance Financial Ratio Reporting. It also gives an illustration of how to use the Model View Controller (MVC) pattern to design a web application. It also shows how to use Agile (SCRUM) for project development and how to use the Codeception framework to conduct unit and acceptance tests. This work can be applied in educating people on how to adopt the MVC architecture, and also on how to prepare Microfinance Financial Ratio reports.

The system can be enhanced in the future to include an interpretation module that is very cohesive to guide the management on what each financial ratio implies through making an in-depth comparison between past financial ratio values and the current one. The system can also be made an expert system that advises the management based on its comparison results.

ACKNOWLEDGEMENT

My acknowledgment goes to the entire staff of Alvana Microfinance Bank Limited for their cooperation, guide and friendly working environment. My acknowledgment also goes to Andela Talent Accelerator for exposing me to the intricacies of automated software application testing, to Twitter for Bootstrap, to Codeception group, to FontAwesome Group, to SpryMedia Limited for the Datatables library and to Olivier Plathey for the FPDF library. Finally, my acknowledgment goes to my family and friends for their encouragement.

REFERENCES

[1] Isabelle Barres et al "Measuring Performance of Microfinance Institutions: A Framework for Reporting, Analysis, and Monitoring" United States Agency for International Development. September 2005.

[2] Patrick E. Akpala "Microfinance Certification Programme Training Manual" Microfinance Certification Programme Committee. 2014.

[3] Nancy Natilson et al "Financial Performance Monitoring: A Guide for Board Members of Microfinance Institutions" Microenterprise Best Practices. Development Alternatives Inc. 7250 Woodmont Avenue Suite 200, Bethesda, MD 20811 USA.

[4] Ruth Dueck Mbeba. "Basic Financial Management and Ratio Analysis for MFIs Toolkit" Mennonite Economic Development Associates (MEDA), http://www.meda.org. March, 2008.

[5] Naftanaila Ionel. "Agile Software Development Methodologies: An Overview of the Current State of Research" University of Economic Studies (ASE) Bucharest Faculty of Management Piata Romana 6, Bucharest, Romania. December, 2009.

[6] Emerson Taymor. "Agile Handbook". http://www.philosophie.is. July, 2016.

APPENDIX – Generate Report File

```
<?php
require_once '../fpdf181/fpdf.php';
require_once '../models/FinancialSelfSufficiency.php';
require_once '../models/OperationalSelfSufficiency.php';
require_once '../models/AdjustedReturnOnAssets.php';
require_once '../models/AdjustedReturnOnEquity.php';
require_once '../models/YieldOnGrossPortfolio.php';
require_once '../models/PortfolioToAsset.php';
require_once '../models/CostOfFunds.php';
require_once '../models/DebtToEquity.php';
require_once '../models/LiquidityRatio.php';
require_once '../models/PortfolioAtRisk.php';
require_once '../models/WriteOffRatio.php';
require_once '../models/CostPerClient.php';
require_once '../models/ManagementReportPeriod.php';

class FinancialRatioGenerator extends fpdf {
  //OSS Variables
  private $financial_revenue, $financial_expense, $loan_los
ses, $operating_expense, $operational_self_sufficiency = 0;
  //FSS Variables
  private $adjusted_financial_expense, $net_loan_losses,$i
nflation_adjustment,$subsidized_cost_of_fund_adjustment,
    $financial_self_sufficiency = 0;
  //AROA Variables
  private $adjusted_net_operating_income, $taxes, $adjuste
d_average_assets, $adjusted_return_on_assets = 0;
  //AROE Variables
  private $adjusted_average_equity, $adjusted_return_on_
equity = 0;
  //Yield on Gross Loan Portfolio Variables
  private $cash_from_gross_loan_portfolio, $average_gross
_loan_portfolio, $yield_on_gross_portfolio = 0;
  //Portfolio To Assets Variables
  private $gross_loan_portfolio, $assets, $portfolio_to_asset
 = 0;
```

```
   //Cost of Funds Variables
   private $financial_expense_on_funding_liabilities, $avera
ge_deposit, $average_borrowings, $cost_of_funds = 0;
   //Debt to Equity Variables
   private $liabilities, $equity, $debt_to_equity = 0;
   //Liquidity Ratio Variables
   private $liquidity_ratio, $cash, $trade_investments, $dem
and_deposits, $short_term_time_deposits, $int_payable_fu
nding_lia, $accounts_payable, $other_current_liabilities = 0;
   // Portfolio at Risk Variable
   private $principal_outstanding_on_all_past_due_loans, $
renegotiated_loans, $portfolio_at_risk = 0;
   //Write-Off Ratio Variables
   private $write_off_ratio, $value_of_loans_written_off = 0;
   //Operating Expense Ratio Variables
   private $operating_expense_ratio = 0;
   //Cost Per Client Variables
   private $cost_per_client_ratio, $average_number_of_clien
ts = 0;

   public $period;
   private $remark = "";
   public function __construct() {
      parent::__construct();
      $this->SetAutoPageBreak(true);
   }
   function pageBorder() {
      $this->SetDrawColor(34,177,76);
      $this->Rect(4,4, 200,290);
   }
   function Header() {
      $this->pageBorder();
   }
   function SetPageHeading() {
      $this->SetFont('Arial','B',24);
      $this-
>Cell(190,10,'ALVANA MICROFINANCE BANK LIMITED'
, 0, 1,"C");
      $this-
>Image("../images/logo square.png",95, 18, 15, 15, "PNG");
      $this->SetFont('Arial','I',10);
      $this-
>Cell(190,30,'Along  Orlu Road, Alvan Ikoku Federal Colleg
e Of Education', 0, 1,"C");
      $this->SetFont('Arial','BU',15);
      $this-
>Cell(190,0,'FINANCIAL RATIO REPORT FOR THE MON
TH OF '.$this->period, 0, 1,"C");
   }
   function SetDocumentInformation() {
      $this->SetCreator("Alvana Microfinance Bank Ltd.");
      $this->SetAuthor("Alvana Microfinance Bank Ltd.");
      $this-
>SetKeywords("Financial Self Sufficiency, Operational Self
Sufficiency, Return On Assets,
         Return On Equity, Sustainability and Profitability, Yi
eld on Gross Portfolio, Portfolio to Asset,
         Cost of Funds, Debt to Equity, Liquidity Ratio, Asset
and Liability Management Ratios,
         Portfolio Quality Ratios, Portfolio at Risk, Write-
off Ratio, Efficiency Ratio,
         Operating Expense Ratio, Cost per Client");
      $this->SetSubject("Financial Ratio Report");
      $this->SetTitle("Financial Ratio Report");
   }
   function ComputeOperationalSelfSufficiency($period_id)
 {
      $operationalSelfSufficiency = new OperationalSelfSuffi
ciency();
      $wherDataArray = array($period_id, 1);
      $wherFieldArray = array("period_id", "status");
      $logic = "AND";
      $operationalSelfSufficiencyRecord = $operationalSelfSu
fficiency->select($wherDataArray,
         $wherFieldArray, $logic);

      //Compute Quantities
      if (@$operationalSelfSufficiencyRecord[0]) {
         $this-
>loan_losses = $operationalSelfSufficiencyRecord[0]['loan_l
osses_expense'];

         $this-
>financial_revenue = $operationalSelfSufficiency-
>computeFinancialRevenue($operationalSelfSufficiencyRec
ord[0]['interest_earned_in_cash'],
         $operationalSelfSufficiencyRecord[0]['income_from_
fees'], $operationalSelfSufficiencyRecord[0]['commissions'],
         $operationalSelfSufficiencyRecord[0]['interest_accru
ed_but_not_yet_earned']);

         $this-
>financial_expense = $operationalSelfSufficiency-
>computeFinancialExpense(
            $operationalSelfSufficiencyRecord[0]['interest_pai
d_in_cash'],
                        $operationalSelfSufficiencyRecord[0]['fee
s_paid'],
                         $operationalSelfSufficiencyRecord[0]['c
ommissions_paid'],
                         $operationalSelfSufficiencyRecord[0]['a
ccrued_interest_but_not_yet_paid']);

         $this-
>operating_expense = $operationalSelfSufficiency-
>computeOperatingExpense(
            $operationalSelfSufficiencyRecord[0]['personnel_e
xpense'],
                        $operationalSelfSufficiencyRecord[0]
['administrative_expense']);

          $this-
>operational_self_sufficiency = $operationalSelfSufficiency-
>computeOperationalSelfSufficiency(
            $this->financial_revenue, $this-
>financial_expense, $this->loan_losses, $this-
>operating_expense);
        }

    }
    function SetOperationalSelfSufficiency($period_id) {
        $this-
>ComputeOperationalSelfSufficiency($period_id);
        //display the computed ratio and it's corresponding val
ues
        $this->SetFont('Arial','',12);
        $this->Cell(10,10,'a.', 0, 0,"J");
        $this->SetFont('Arial','',12);
        $this-
>Cell(10,10,'Operational Self Sufficiency (OSS) = ', 0, 1,"J");

        $this->SetFont('Arial','U',12);
        //spaces are to create a division effect
        $this-
>Cell(150,5,'                    Financial Revenue
      ', 0, 1,"C");

        $this->SetFont('Arial','',12);
        $this-
>Cell(150,10,'(Financial Expense + Loan Losses + Operating
Expenses)', 0, 1,"C");

        $this->Cell(150,5,'=', 0, 1,"C");

        $this->SetFont('Arial','U',12);
        $this->Cell(150,10,"          ".$this-
>financial_revenue."           ", 0, 1,"C");
        $this->SetFont('Arial','',12);
        $this->Cell(150,5,"(".$this->financial_expense."+".$this-
>loan_losses.
        "+".$this->operating_expense.")", 0, 1,"C");
        $this->Cell(150,5,'=', 0, 1,"C");
        $this->SetFont('Arial','U',12);
        $this->Cell(150,10,"          ".$this-
>financial_revenue."           ", 0, 1,"C");
        $this->SetFont('Arial','',12);
        $this->Cell(150, 5, $this->financial_expense + $this-
>loan_losses+
        $this->operating_expense, 0, 1,"C");
        $this->Cell(150,5,'=', 0, 1,"C");
        $this->SetFont('Arial','',12);
        $denominator = ($this->financial_expense + $this-
>loan_losses+
        $this->operating_expense);
        if ($denominator != 0)
           $this->operational_self_sufficiency = $this-
>financial_revenue/$denominator;
        else
           $this->operational_self_sufficiency = 0;

        if ($this->operational_self_sufficiency < 100) {
           $this->SetTextColor(237,28,36);
           $this-
>remark = "The Bank is not operationally self sufficient, try
getting your operational self sufficiency to 100% or above 1
00%.";
        }
        else {
           $this->SetTextColor(34,177,76);
           $this-
>remark = "The Bank is operationally self sufficient, try kee
ping your operational self sufficiency to 100% or above 100
%.";
        }
        $this->Cell(150, 5,$this-
>operational_self_sufficiency , 0, 1,"C");
        $this->Write(6, "NOTE: ".$this->remark);
    }

    function ComputeFinancialSelfSufficiency($period_id) {
        $financialSelfSufficiency = new FinancialSelfSufficiency
();
        $wherDataArray = array($period_id, 1);
        $wherFieldArray = array("period_id", "status");
        $logic = "AND";
        $financialSelfSufficiencyRecord = $financialSelfSufficie
ncy->select($wherDataArray,
           $wherFieldArray, $logic);

        //Compute Quantities
        if (@$financialSelfSufficiencyRecord[0]) {
           $this-
>inflation_adjustment = $financialSelfSufficiency-
>computeInflationAdjustment(
              $financialSelfSufficiencyRecord[0]['average_equity
'],
              $financialSelfSufficiencyRecord[0]['average_fixed_
assets'],
              $financialSelfSufficiencyRecord[0]['inflation_rate'])
;

           $this-
```

```
>subsidized_cost_of_fund_adjustment = $financialSelfSuffic
iency->computeSubsidizedCostOfFundAdjustment(
        $financialSelfSufficiencyRecord[0]['average_fundi
ng_liabilities'], $financialSelfSufficiencyRecord[0]['commerc
ial_rate_for_funds'],
        $financialSelfSufficiencyRecord[0]['interest_and_fe
es_expense']);

      $this-
>adjusted_financial_expense = $financialSelfSufficiency-
>computeAdjustedFinancialExpense(
        $this->inflation_adjustment, $this-
>subsidized_cost_of_fund_adjustment);

      $this->net_loan_losses = $financialSelfSufficiency-
>computeNetLoanLosses($financialSelfSufficiencyRecord[0
]['gross_loan_losses'],
      $financialSelfSufficiencyRecord[0]['lost_interest_ded
uctions'] );

      $this-
>financial_self_sufficiency = $financialSelfSufficiency-
>ComputeFinancialSelfSufficiency(
        $this->financial_revenue, $this-
>adjusted_financial_expense, $this->net_loan_losses,
        $this->operating_expense );
    }
  }
  function SetFinancialSelfSufficiency($period_id) {
    $this->ComputeFinancialSelfSufficiency($period_id);

    $this->SetTextColor(0,0,0);

    //display the computed ratio and it's corresponding val
ues
    $this->SetFont('Arial','',12);
    $this->ln();
    $this->Cell(5,10,'b.', 0, 0,"J");
    $this->SetFont('Arial','',12);
    $this-
>Cell(5,10,'Financial Self Sufficiency (FSS) = ', 0, 1,"J");

    $this->SetFont('Arial','U',12);
    //spaces are to create a division effect
    $this-
>Cell(150,5,'                  Financial Revenue
   ', 0, 1,"C");

    $this->SetFont('Arial','',12);
    $this-
>Cell(150,5,'(Adjusted Financial Expense + Net Loan Losses
+ Operating Expenses)', 0, 1,"C");
```

```
    $this->Cell(150,5,'=', 0, 1,"C");

    $this->SetFont('Arial','U',12);
    $this->Cell(150,5,"          ".$this-
>financial_revenue."          ", 0, 1,"C");
    $this->SetFont('Arial','',12);
    $this->Cell(150,5,"(".$this-
>adjusted_financial_expense."+".$this->net_loan_losses."+".
    $this->operating_expense.")", 0, 1,"C");
    $this->Cell(150,5,'=', 0, 1,"C");
    $this->SetFont('Arial','U',12);
    $this->Cell(150,5,"          ".$this-
>financial_revenue."          ", 0, 1,"C");
    $this->SetFont('Arial','',12);
    $this->Cell(150, 5, $this-
>adjusted_financial_expense + $this->net_loan_losses +
    $this->operating_expense, 0, 1,"C");
    $this->Cell(150,5,'=', 0, 1,"C");
    $this->SetFont('Arial','',12);

    if ($this->financial_self_sufficiency < 100) {
      $this->SetTextColor(237,28,36);
      $this-
>remark = "The Bank is not financially self sufficient, try get
ting your financial self sufficiency to 100% or above 100%.";
    }
    else {
      $this->SetTextColor(34,177,76);
      $this-
>remark = "The Bank is financially self sufficient, try keepin
g your financial self sufficiency at 100% or above 100%. Als
o note that an increasing FSS is positive.";
    }
    $this->Cell(150, 5,$this-
>financial_self_sufficiency , 0, 1,"C");
    $this->Write(6, "NOTE: ".$this->remark);
  }
  function ComputeAdjustedReturnOnAssets($period_id) {
    $adjustedReturnOnAssets = new AdjustedReturnOnAs
sets();
    $wherDataArray = array($period_id, 1);
    $wherFieldArray = array("period_id", "status");
    $logic = "AND";
    $adjustedReturnOnAssetsRecord = $adjustedReturnOn
Assets->select($wherDataArray,
      $wherFieldArray, $logic);

    //Compute Quantities
    if (@$adjustedReturnOnAssetsRecord[0]) {
      $this-
>adjusted_net_operating_income = $adjustedReturnOnAsse
tsRecord[0]['adjusted_net_operating_income'];
      $this-
>taxes = $adjustedReturnOnAssetsRecord[0]['taxes'];
          $this-
>adjusted_average_assets = $adjustedReturnOnAssetsRecor
d[0]['adjusted_average_assets'];
          if ($this->adjusted_average_assets)
             $this-
>adjusted_return_on_assets = $adjustedReturnOnAssets-
>computeAdjustedReturnOnAssets(
                $this->adjusted_net_operating_income, $this-
>taxes, $this->adjusted_average_assets);
          else
             $this->adjusted_return_on_assets = 0;
       }
    }
    function SetAdjustedReturnOnAssets($period_id) {
       $this->ComputeAdjustedReturnOnAssets($period_id);
       $this->SetTextColor(0,0,0);

       //display the computed ratio and it's corresponding val
ues
       $this->SetFont('Arial','',12);
       $this->ln();
       $this->Cell(10,20,'c.', 0, 0,"J");
       $this->SetFont('Arial','',12);
       $this-
>Cell(10,20,'Adjusted Return On Assets (AROA) = ', 0, 1,"J");

       $this->SetFont('Arial','U',12);
       //spaces are to create a division effect
       $this-
>Cell(150,5,'    Adjusted Net Operating Income - Taxes    ', 0,
1,"C");

       $this->SetFont('Arial','',12);
       $this->Cell(150,10,'Adjusted Average Assets', 0, 1,"C");

       $this->Cell(150,5,'=', 0, 1,"C");

       $this->SetFont('Arial','U',12);
       $this->Cell(150,10,$this-
>adjusted_net_operating_income." - ".$this->taxes, 0, 1,"C");
       $this->SetFont('Arial','',12);
       $this->Cell(150,5, $this-
>adjusted_average_assets, 0, 1,"C");
       $this->Cell(150,5,'=', 0, 1,"C");
       $this->SetFont('Arial','U',12);
       $this->Cell(150,10,($this-
>adjusted_net_operating_income - $this->taxes), 0, 1,"C");
       $this->SetFont('Arial','',12);
       $this->Cell(150, 5, $this-
>adjusted_average_assets, 0, 1,"C");
       $this->Cell(150,5,'=', 0, 1,"C");
       $this->SetFont('Arial','',12);
```

```
       $this->Cell(150, 5,$this-
>adjusted_return_on_assets , 0, 1,"C");
       $this->SetTextColor(34,177,76);
       $this->remark = "An increasing AROA is positive.";
       $this->Write(6, "NOTE: ".$this->remark);

    }
    function ComputeAdjustedReturnOnEquity($period_id)
{
       $adjustedReturnOnEquity = new AdjustedReturnOnEq
uity();
       $wherDataArray = array($period_id, 1);
       $wherFieldArray = array("period_id", "status");
       $logic = "AND";
       $adjustedReturnOnEquityRecord = $adjustedReturnOn
Equity->select($wherDataArray,
          $wherFieldArray, $logic);

       //Compute Quantities
       if (@$adjustedReturnOnEquityRecord[0] && $this-
>adjusted_net_operating_income && $this->taxes) {
          $this-
>adjusted_average_equity = $adjustedReturnOnEquityReco
rd[0]['adjusted_average_equity'];
             $this-
>adjusted_return_on_equity = $adjustedReturnOnEquity-
>computeAdjustedReturnOnEquity($this-
>adjusted_net_operating_income,
             $this->taxes, $this->adjusted_average_equity);
       }
    }
    function SetAdjustedReturnOnEquity($period_id) {
       $this->ComputeAdjustedReturnOnEquity($period_id);
       $this->SetTextColor(0,0,0);

       //display the computed ratio and it's corresponding val
ues
       $this->SetFont('Arial','',12);
       $this->ln();
       $this->Cell(10,20,'d.', 0, 0,"J");
       $this->SetFont('Arial','',12);
       $this-
>Cell(10,20,'Adjusted Return On Equity (AROE) = ', 0, 1,"J");

       $this->SetFont('Arial','U',12);
       //spaces are to create a division effect
       $this-
>Cell(150,5,'    Adjusted Net Operating Income - Taxes    ', 0
, 1,"C");

       $this->SetFont('Arial','',12);
       $this->Cell(150,10,'Adjusted Average Equity', 0, 1,"C");
```

```
        $this->Cell(150,5,'=', 0, 1,"C");

        $this->SetFont('Arial','U',12);
        $this->Cell(150,10,$this-
>adjusted_net_operating_income." - ".$this->taxes, 0, 1,"C");
        $this->SetFont('Arial','',12);
        $this->Cell(150,5, $this-
>adjusted_average_equity, 0, 1,"C");
        $this->Cell(150,5,'=', 0, 1,"C");
        $this->SetFont('Arial','U',12);
        $this->Cell(150,10,($this-
>adjusted_net_operating_income - $this->taxes), 0, 1,"C");
        $this->SetFont('Arial','',12);
        $this->Cell(150, 5, $this-
>adjusted_average_equity, 0, 1,"C");
        $this->Cell(150,5,'=', 0, 1,"C");
        $this->SetFont('Arial','',12);
        $this->Cell(150, 5,$this-
>adjusted_return_on_equity , 0, 1,"C");
        $this->SetTextColor(34,177,76);
        $this->remark = "An increasing AROE is positive.";
        $this->Write(6, "NOTE: ".$this->remark);

    }
    function ComputeYieldOnGrossLoanPortfolio($period_i
d) {
        $yieldOnGrossPortfolio = new YieldOnGrossPortfolio()
;
        $wherDataArray = array($period_id, 1);
        $wherFieldArray = array("period_id", "status");
        $logic = "AND";
        $yieldOnGrossPortfolioRecord = $yieldOnGrossPortfol
io->select($wherDataArray,
            $wherFieldArray, $logic);

        //Compute Quantities
        if (@$yieldOnGrossPortfolioRecord[0]) {
            $this-
>cash_from_gross_loan_portfolio = $yieldOnGrossPortfolio
Record[0]['cash_from_gross_loan_portfolio'];
            $this-
>average_gross_loan_portfolio = $yieldOnGrossPortfolioRe
cord[0]['average_gross_loan_portfolio'];

            $this-
>yield_on_gross_portfolio = $yieldOnGrossPortfolio-
>computeYieldOnGrossPortfolio(
                    $this->cash_from_gross_loan_portfolio, $this-
>average_gross_loan_portfolio);
        }
    }
    function SetYieldOnGrossLoanPortfolio($period_id) {
        $this-
>ComputeYieldOnGrossLoanPortfolio($period_id);
        $this->SetTextColor(0,0,0);

        //display the computed ratio and it's corresponding val
ues
        $this->SetFont('Arial','',12);
        $this->Cell(10,5,'a.', 0, 0,"J");
        $this->SetFont('Arial','',12);
        $this-
>Cell(10,5,'Yield on Gross Portfolio or Portfolio Yield  = ', 0,
1,"J");

        $this->SetFont('Arial','U',12);
        //spaces are to create a division effect
        $this-
>Cell(150,10,'Cash received from interest,fees and commisio
ns on Loan Portfolio', 0, 1,"C");

        $this->SetFont('Arial','',12);
        $this-
>Cell(150,5,'Average Gross Loan Portfolio', 0, 1,"C");

        $this->Cell(150,5,'=', 0, 1,"C");

        $this->SetFont('Arial','U',12);
        $this->Cell(150,10,$this-
>cash_from_gross_loan_portfolio, 0, 1,"C");
        $this->SetFont('Arial','',12);
        $this->Cell(150,5, $this-
>average_gross_loan_portfolio, 0, 1,"C");
        $this->Cell(150,5,'=', 0, 1,"C");
        $this->SetFont('Arial','',12);
        $this->Cell(150, 5,$this-
>yield_on_gross_portfolio , 0, 1,"C");
        $this->SetTextColor(34,177,76);
        $this-
>remark = "Your Portfolio Yield should be compared agains
t effective interest rate of loans; if your yield is significantly/
consistently lower than the effective interest rate, it means y
our MFI has a problem with Loan collections. Also, an incre
asing yield is positive although it will level off as it nears th
e effective interest rate.";
        $this->Write(6, "NOTE: ".$this->remark);

    }
    function ComputePortfolioToAsset($period_id) {
        $portfolioToAsset = new PortfolioToAsset();
        $wherDataArray = array($period_id, 1);
        $wherFieldArray = array("period_id", "status");
        $logic = "AND";
        $portfolioToAssetRecord = $portfolioToAsset-
>select($wherDataArray,
            $wherFieldArray, $logic);
    //Compute Quantities
    if (@$portfolioToAssetRecord[0]) {
      $this-
>gross_loan_portfolio = $portfolioToAssetRecord[0]['gross_
loan_portfolio'];
      $this->assets = $portfolioToAssetRecord[0]['assets'];
      $this->portfolio_to_asset = $portfolioToAsset-
>computePortfolioToAsset($this->gross_loan_portfolio,
        $this->assets);
    }
  }
  function SetPortfolioToAsset($period_id) {
    $this->ComputePortfolioToAsset($period_id);
    $this->SetTextColor(0,0,0);
    $this->ln();

    //display the computed ratio and it's corresponding val
ues
    $this->SetFont('Arial','',12);
    $this->Cell(10,5,'b.', 0, 0,"J");
    $this->SetFont('Arial','',12);
    $this->Cell(10,5,'Portfolio to Assets = ', 0, 1,"J");

    $this->SetFont('Arial','U',12);
    //spaces are to create a division effect
    $this->Cell(150,10,'Gross Loan Portfolio', 0, 1,"C");

    $this->SetFont('Arial','',12);
    $this->Cell(150,5,'Assets', 0, 1,"C");

    $this->Cell(150,5,'=', 0, 1,"C");

    $this->SetFont('Arial','U',12);
    $this->Cell(150,10,$this->gross_loan_portfolio, 0, 1,"C");
    $this->SetFont('Arial','',12);
    $this->Cell(150,5, $this->assets, 0, 1,"C");
    $this->Cell(150,5,'=', 0, 1,"C");
    $this->SetFont('Arial','',12);
    $this->Cell(150, 5,$this->portfolio_to_asset , 0, 1,"C");
    $this->SetTextColor(34,177,76);
    $this-
>remark = "Your Portfolio to Assets indicates how well your
 MFI is allocating her assets to granting of loans to micro-
entrepreneurs. Also, an increasing trend is positive.";
    $this->Write(6, "NOTE: ".$this->remark);
  }
  function ComputeCostOfFunds($period_id) {
    $costOfFunds = new CostOfFunds();
    $wherDataArray = array($period_id, 1);
    $wherFieldArray = array("period_id", "status");
    $logic = "AND";
    $costOfFundsRecord = $costOfFunds-
>select($wherDataArray,
      $wherFieldArray, $logic);

    //Compute Quantities
    if (@$costOfFundsRecord[0]) {
      $this-
>financial_expense_on_funding_liabilities = $costOfFundsR
ecord[0]['financial_expense_on_funding_liabilities'];
      $this-
>average_deposit = $costOfFundsRecord[0]['average_depos
it'];
      $this-
>average_borrowings = $costOfFundsRecord[0]['average_b
orrowings'];

      $this->cost_of_funds = $costOfFunds-
>computeCostOfFunds($this-
>financial_expense_on_funding_liabilities,
        $this->average_deposit, $this-
>average_borrowings);
    }
  }
  function SetCostOfFunds($period_id) {
    $this->ComputeCostOfFunds($period_id);
    $this->SetTextColor(0,0,0);
    $this->ln();
    $this->ln();
    //display the computed ratio and it's corresponding val
ues
    $this->SetFont('Arial','',12);
    $this->Cell(10,5,'c.', 0, 0,"J");
    $this->SetFont('Arial','',12);
    $this->Cell(10,5,'Cost of Funds = ', 0, 1,"J");

    $this->SetFont('Arial','U',12);
    //spaces are to create a division effect
    $this-
>Cell(150,10,'Financial Expense on Funding Liabilities', 0, 1,
"C");

    $this->SetFont('Arial','',12);
    $this-
>Cell(150,5,'(Average Deposit + Average Borrowings)', 0, 1,"
C");

    $this->Cell(150,5,'=', 0, 1,"C");

    $this->SetFont('Arial','U',12);
    $this->Cell(150,10,"          ".$this-
>financial_expense_on_funding_liabilities."          ", 0, 1,"C");
    $this->SetFont('Arial','',12);
    $this->Cell(150,5, $this->average_deposit." + ".$this-
>average_borrowings, 0, 1,"C");
```

```
        $this->Cell(150,5,'=', 0, 1,"C");
        $this->SetFont('Arial','U',12);
        $this->Cell(150,10,"   ".$this-
>financial_expense_on_funding_liabilities."    ", 0, 1,"C");
        $this->SetFont('Arial','',12);
        $this->Cell(150,5, ($this->average_deposit + $this-
>average_borrowings), 0, 1,"C");
        $this->Cell(150,5,'=', 0, 1,"C");
        $this->SetFont('Arial','',12);
        $this->Cell(150, 5,$this->cost_of_funds , 0, 1,"C");
        $this->SetTextColor(34,177,76);
        $this-
>remark = "A decreasing Cost of Funds ratio is generally po
sitive. For a successful interest rate management determinat
ion, the portfolio yield is compared to the cost of funding th
e gross loan portfolio with borrowings. Cost of fund should
 always be less than the portfolio yield. Efforts should be m
ade to minimize cost of funds and maximize portfolio yield.
 Comparing the cost of funds to the portfolio yield gives the
 MFI financial spread";
        $this->Write(6, "NOTE: ".$this->remark);
    }
    function ComputeDebtToEquity($period_id) {
        $debtToEquity = new DebtToEquity();
        $wherDataArray = array($period_id, 1);
        $wherFieldArray = array("period_id", "status");
        $logic = "AND";
        $debtToEquityRecord = $debtToEquity-
>select($wherDataArray,
            $wherFieldArray, $logic);

        //Compute Quantities
        if (@$debtToEquityRecord[0]) {
            $this-
>liabilities = $debtToEquityRecord[0]['liabilities'];
            $this->equity = $debtToEquityRecord[0]['equity'];
            $this->debt_to_equity = $debtToEquity-
>computeDebtToEquity($this->liabilities, $this->equity);
        }
    }
    function SetDebtToEquityRatio($period_id) {
        $this->ComputeDebtToEquity($period_id);
        $this->SetTextColor(0,0,0);
        $this->ln();
        $this->ln();
        //display the computed ratio and it's corresponding val
ues
        $this->SetFont('Arial','',12);
        $this->Cell(10,5,'d.', 0, 0,"J");
        $this->SetFont('Arial','',12);
        $this-
>Cell(10,5,'Debt to Equity / Leverage Ratio = ', 0, 1,"J");
        $this->SetFont('Arial','U',12);
        //spaces are to create a division effect
        $this->Cell(150,10,'Liabilties', 0, 1,"C");

        $this->SetFont('Arial','',12);
        $this->Cell(150,5,'Equity', 0, 1,"C");

        $this->Cell(150,5,'=', 0, 1,"C");

        $this->SetFont('Arial','U',12);
        $this->Cell(150,10,$this->liabilities, 0, 1,"C");
        $this->SetFont('Arial','',12);
        $this->Cell(150,5, $this->equity, 0, 1,"C");
        $this->Cell(150,5,'=', 0, 1,"C");
        $this->Cell(150, 5,$this->debt_to_equity , 0, 1,"C");
        if ($this->debt_to_equity <= 0.5) {
            $this->SetTextColor(34,177,76);
            $this-
>remark = "Your debt to Equity ratio is ok. This ratio shows
safety cushion the bank has to absorb losses before creditors
 are at risk. It also shows how well the MFB is able to levera
ge its Equity to increase assets through borrowing. It is advi
sable not to have a leverage ratio that is not more than 1:2.";
        }
        else {
            $this->SetTextColor(237,28,36);
            $this-
>remark = "Your debt to Equity ratio is not ok. This ratio sh
ows safety cushion the bank has to absorb losses before cre
ditors are at risk. It also shows how well the MFB is able to l
everage its Equity to increase assets through borrowing. It i
s advisable not to have a leverage ratio that is not more than
 1:2.";
        }
        $this->Write(6, "NOTE: ".$this->remark);
    }
    function ComputeLiquidityRatio($period_id) {
        $liquidityRatio = new LiquidityRatio();
        $wherDataArray = array($period_id, 1);
        $wherFieldArray = array("period_id", "status");
        $logic = "AND";
        $liquidityRatioRecord = $liquidityRatio-
>select($wherDataArray,
            $wherFieldArray, $logic);

        //Compute Quantities
        if (@$liquidityRatioRecord[0]) {
            $this->cash = $liquidityRatioRecord[0]['cash'];
            $this-
>trade_investments = $liquidityRatioRecord[0]['trade_inves
tments'];
            $this-
>demand_deposits = $liquidityRatioRecord[0]['demand_de
```

```
posits'];
      $this-
>short_term_time_deposits = $liquidityRatioRecord[0]['shor
t_term_time_deposits'];
      $this-
>int_payable_funding_lia = $liquidityRatioRecord[0]['int_p
ayable_funding_lia'];
      $this-
>accounts_payable = $liquidityRatioRecord[0]['accounts_pa
yable'];
      $this-
>other_current_liabilities = $liquidityRatioRecord[0]['other_
current_liabilities'];

      $this->liquidity_ratio = $liquidityRatio-
>computeLiquidityRatio($this->cash, $this-
>trade_investments,
      $this->demand_deposits, $this-
>short_term_time_deposits, $this-
>int_payable_funding_lia, $this->accounts_payable,
      $this->other_current_liabilities);
    }
  }
  function SetLiquidityRatio($period_id) {
    $this->ComputeLiquidityRatio($period_id);
    $this->SetTextColor(0,0,0);
    $this->ln();
    $this->ln();
    //display the computed ratio and it's corresponding val
ues
    $this->SetFont('Arial','',12);
    $this->Cell(10,5,'e.', 0, 0,"J");
    $this->SetFont('Arial','',12);
    $this->Cell(10,5,'Liquidity Ratio = ', 0, 1,"J");

    $this->SetFont('Arial','U',9);
    //spaces are to create a division effect
    $this-
>Cell(150,10,'          Cash + Trade Investments          ', 0, 1,"
C");

    $this->SetFont('Arial','',9);
    $this->Cell(185,5,'(Demand Deposits + Short-
term Time Deposits + Interest Payable on Funding Liabilitie
s + Accounts Payable + Other Current Liabilities)', 0, 1,"C");

    $this->Cell(150,5,'=', 0, 1,"C");

    $this->SetFont('Arial','U',12);
    $this->Cell(150,10,"          ".$this->cash." + ".$this-
>trade_investments."          ", 0, 1,"C");
    $this->SetFont('Arial','',12);
    $this->Cell(150,5, "( ".$this-
```

```
>demand_deposits." + ".$this-
>short_term_time_deposits." + ".$this-
>int_payable_funding_lia." + ".$this-
>accounts_payable." + ".$this-
>other_current_liabilities." )", 0, 1,"C");
    $this->Cell(150,5,'=', 0, 1,"C");

    $this->SetFont('Arial','U',12);
    $this->Cell(150,10,($this->cash + $this-
>trade_investments), 0, 1,"C");
    $this->SetFont('Arial','',12);
    $this->Cell(150,5, ($this->demand_deposits + $this-
>short_term_time_deposits + $this-
>int_payable_funding_lia + $this->accounts_payable + $this-
>other_current_liabilities), 0, 1,"C");
    $this->Cell(150,5,'=', 0, 1,"C");
    $this->Cell(150, 5,$this->liquidity_ratio , 0, 1,"C");

    $this->SetTextColor(34,177,76);
    $this-
>remark = "The liquidity Ratio is a measurement of the suffi
ciency of cash resources to pay the short-
term obligations to depositors, lenders and other creditors."
;

    $this->Write(6, "NOTE: ".$this->remark);
  }
  function ComputePortfolioAtRiskRatio($period_id) {
    $portfolioAtRisk = new PortfolioAtRisk();
    $wherDataArray = array($period_id, 1);
    $wherFieldArray = array("period_id", "status");
    $logic = "AND";
    $portfolioAtRiskRecord = $portfolioAtRisk-
>select($wherDataArray,
      $wherFieldArray, $logic);

    //Compute Quantities
    if (@$portfolioAtRiskRecord[0]) {
      $this-
>principal_outstanding_on_all_past_due_loans = $portfolio
AtRiskRecord[0]['principal_outstanding_on_all_past_due_l
oans'];
      $this-
>renegotiated_loans = $portfolioAtRiskRecord[0]['renegotia
ted_loans'];
      $this->portfolio_at_risk = $portfolioAtRisk-
>computePortfolioAtRisk($this-
>principal_outstanding_on_all_past_due_loans,
      $this->renegotiated_loans, $this-
>gross_loan_portfolio);
    }
  }
  function SetPortfolioAtRisk($period_id) {
        $this->ComputePortfolioAtRiskRatio($period_id);
        $this->SetTextColor(0,0,0);

        //display the computed ratio and it's corresponding val
ues
        $this->SetFont('Arial','',12);
        $this->Cell(10,5,'a.', 0, 0,"J");
        $this->SetFont('Arial','',12);
        $this->Cell(10,5,'Portfolio at Risk (PAR) = ', 0, 1,"J");

        $this->SetFont('Arial','U',12);
        //spaces are to create a division effect
        $this-
>Cell(150,10,'Sum of principal outstanding on all past-
due loans + Renegotiated Loans', 0, 1,"C");

        $this->SetFont('Arial','',12);
        $this->Cell(150,5,'Gross Loan Portfolio', 0, 1,"C");

        $this->Cell(150,5,'=', 0, 1,"C");

        $this->SetFont('Arial','U',12);
        $this->Cell(150,10,"     ".$this-
>principal_outstanding_on_all_past_due_loans." + ".$this-
>renegotiated_loans."       ", 0, 1,"C");
        $this->SetFont('Arial','',12);
        $this->Cell(150,5, $this->gross_loan_portfolio, 0, 1,"C");
        $this->Cell(150,5,'=', 0, 1,"C");

        $this->SetFont('Arial','U',12);
        $this->Cell(150,10,($this-
>principal_outstanding_on_all_past_due_loans + $this-
>renegotiated_loans), 0, 1,"C");
        $this->SetFont('Arial','',12);
        $this->Cell(150,5, $this->gross_loan_portfolio, 0, 1,"C");
        $this->Cell(150,5,'=', 0, 1,"C");
        $this->Cell(150, 5,$this->portfolio_at_risk , 0, 1,"C");

        if ($this->portfolio_at_risk <= 2.5) {
          $this->SetTextColor(34,177,76);
          $this-
>remark = "Your Portfolio at Risk ratio is ok. A decreasing P
ortfolio at Risk is positive.";
        }
        else {
          $this->SetTextColor(237,28,36);
          $this-
>remark = "Your Portfolio at Risk ratio is not ok. Best practic
e and regulatory threshold for Nigeria requires that the Port
folio at Risk for MFBs should not exceed 2.5%.";
        }
        $this->Write(6, "NOTE: ".$this->remark);
    }
```

```
    function ComputeWriteOffRatio($period_id) {
      $writeOffRatio = new WriteOffRatio();
      $wherDataArray = array($period_id, 1);
      $wherFieldArray = array("period_id", "status");
      $logic = "AND";
      $writeOffRatioRecord = $writeOffRatio-
>select($wherDataArray,
        $wherFieldArray, $logic);

      //Compute Quantities
      if (@$writeOffRatioRecord[0]) {
        $this-
>value_of_loans_written_off = $writeOffRatioRecord[0]['val
ue_of_loans_written_off'];

        $this->write_off_ratio = $writeOffRatio-
>computeWriteOffRatio($this->value_of_loans_written_off,
          $this->average_gross_loan_portfolio);
      }
    }
    function SetWriteOffRatio($period_id) {
      $this->ComputeWriteOffRatio($period_id);
      $this->SetTextColor(0,0,0);
      $this->ln();

      //display the computed ratio and it's corresponding val
ues
      $this->SetFont('Arial','',12);
      $this->Cell(10,5,'b.', 0, 0,"J");
      $this->SetFont('Arial','',12);
      $this->Cell(10,5,'Write-Off Ratio = ', 0, 1,"J");

      $this->SetFont('Arial','U',12);
      //spaces are to create a division effect
      $this-
>Cell(150,10,'        Value of Loans Written off        ', 0, 1,"C");

      $this->SetFont('Arial','',12);
      $this-
>Cell(150,5,'Average Gross Loan Portfolio', 0, 1,"C");

      $this->Cell(150,5,'=', 0, 1,"C");

      $this->SetFont('Arial','U',12);
      $this->Cell(150,10,$this-
>value_of_loans_written_off, 0, 1,"C");
      $this->SetFont('Arial','',12);
      $this->Cell(150,5, $this-
>average_gross_loan_portfolio, 0, 1,"C");
      $this->Cell(150,5,'=', 0, 1,"C");
      $this->Cell(150, 5,$this->write_off_ratio , 0, 1,"C");

      $this->SetTextColor(34,177,76);
    $this->remark = "A decreasing Write-
off Ratio is positive.";

    $this->Write(6, "NOTE: ".$this->remark);
  }
  function ComputeOperatingExpenseRatio() {
    if ($this->average_gross_loan_portfolio > 0)
      $this->operating_expense_ratio = $this-
>operating_expense / $this->average_gross_loan_portfolio;
    else
      $this->operating_expense_ratio = 0;
  }
  function SetOperatingExpenseRatio() {
    $this->ComputeOperatingExpenseRatio();
    $this->SetTextColor(0,0,0);

    //display the computed ratio and it's corresponding val
ues
    $this->SetFont('Arial','',12);
    $this->Cell(10,5,'a.', 0, 0,"J");
    $this->SetFont('Arial','',12);
    $this->Cell(10,5,'Operating Expense Ratio = ', 0, 1,"J");

    $this->SetFont('Arial','U',12);
    //spaces are to create a division effect
    $this-
>Cell(150,10,'        Operating Expense       ', 0, 1,"C");

    $this->SetFont('Arial','',12);
    $this-
>Cell(150,5,'Average Gross Loan Portfolio', 0, 1,"C");

    $this->Cell(150,5,'=', 0, 1,"C");

    $this->SetFont('Arial','U',12);
    $this->Cell(150,10,$this->operating_expense, 0, 1,"C");
    $this->SetFont('Arial','',12);
    $this->Cell(150,5, $this-
>average_gross_loan_portfolio, 0, 1,"C");
    $this->Cell(150,5,'=', 0, 1,"C");
    $this->Cell(150, 5,$this-
>operating_expense_ratio , 0, 1,"C");

    $this->SetTextColor(34,177,76);
    $this-
>remark = "The Operating Expense Ratio enables managers
to quickly compare administrative and personnel expenses
to the MFB yield on the Gross Loan Portfolio. The lower the
 operating expense ratio, the more efficient the MFB is.";

    $this->Write(6, "NOTE: ".$this->remark);
  }
  function ComputeCostPerClientRatio($period_id) {
    $costPerClient = new CostPerClient();
    $wherDataArray = array($period_id, 1);
    $wherFieldArray = array("period_id", "status");
    $logic = "AND";
    $costPerClientRecord = $costPerClient-
>select($wherDataArray,
      $wherFieldArray, $logic);

    //Compute Quantities
    if (@$costPerClientRecord[0]) {
      $this-
>average_number_of_clients = $costPerClientRecord[0]['ave
rage_number_of_clients'];
      $this->cost_per_client_ratio = $costPerClient-
>computeCostPerClient($this->operating_expense,
        $this->average_number_of_clients);
    }
  }
  function SetCostPerClient($period_id) {
    $this->ComputeCostPerClientRatio($period_id);
    $this->SetTextColor(0,0,0);
    $this->ln();
    //display the computed ratio and it's corresponding val
ues
    $this->SetFont('Arial','',12);
    $this->Cell(10,5,'b.', 0, 0,"J");
    $this->SetFont('Arial','',12);
    $this->Cell(10,5,'Cost per Client Ratio = ', 0, 1,"J");

    $this->SetFont('Arial','U',12);
    //spaces are to create a division effect
    $this-
>Cell(150,10,'        Operating Expense       ', 0, 1,"C");

    $this->SetFont('Arial','',12);
    $this->Cell(150,5,'Average Number of Clients', 0, 1,"C");

    $this->Cell(150,5,'=', 0, 1,"C");

    $this->SetFont('Arial','U',12);
    $this->Cell(150,10,$this->operating_expense, 0, 1,"C");
    $this->SetFont('Arial','',12);
    $this->Cell(150,5, $this-
>average_number_of_clients, 0, 1,"C");
    $this->Cell(150,5,'=', 0, 1,"C");
    $this->Cell(150, 5,$this->cost_per_client_ratio , 0, 1,"C");

    $this->SetTextColor(34,177,76);
    $this-
>remark = "The Cost per Client Ratio indicates to an institut
ion how much it currently spends in Personnel and Admini
strative Expenses to serve a single active client. It informs th
e MFI how much it must earn on average from each client t
```

```
o be profitable.";

        $this->Write(6, "NOTE: ".$this->remark);
    }
    function SetSustainabiltyAndProfitabilityRatios($period_id) {
        $this->SetFont('Arial','BU',14);
        $this->Cell(190,20,'Sustainability and Profitability', 0, 1,"C");
        $this->SetOperationalSelfSufficiency($period_id);
        $this->SetFinancialSelfSufficiency($period_id);
        $this->SetAdjustedReturnOnAssets($period_id);
        $this->SetAdjustedReturnOnEquity($period_id);
    }
    function SetAssetAndLiabilityManagementRatios($period_id) {
        $this->SetFont('Arial','BU',14);
        $this->SetTextColor(0,0,0);
        $this->Cell(190,20,'Asset And Liability Management', 0, 1,"C");
        $this->SetYieldOnGrossLoanPortfolio($period_id);
        $this->SetPortfolioToAsset($period_id);
        $this->SetCostOfFunds($period_id);
        $this->SetDebtToEquityRatio($period_id);
        $this->SetLiquidityRatio($period_id);
    }
    function SetPortfolioQualityRatios($period_id) {
        $this->SetFont('Arial','BU',14);
        $this->SetTextColor(0,0,0);
        $this->Cell(190,20,'Portfolio Quality Ratio', 0, 1,"C");
        $this->SetPortfolioAtRisk($period_id);
        $this->SetWriteOffRatio($period_id);
    }
    function SetEfficiencyRatios($period_id) {
        $this->SetFont('Arial','BU',14);
        $this->SetTextColor(0,0,0);
        $this->Cell(190,20,'Efficiency/Productivity Ratios', 0, 1,"C");
        $this->SetOperatingExpenseRatio();
        $this->SetCostPerClient($period_id);
    }
}

$financialRatioGenerator = new FinancialRatioGenerator();

//get the period that report should be generated for
$period_id = $_GET['period_id'];
$managementReportPeriod = new ManagementReportPeriod();
$whereDataArray = array($period_id);
$whereFieldArray = array("id");
$logic = "";
$managementReportPeriodRecords = $managementReportPeriod->select($whereDataArray,
                                        $whereFieldArray, $logic);
$period = $managementReportPeriodRecords[0]['from_date']." ". $managementReportPeriodRecords[0]['to_date'];

$financialRatioGenerator = new FinancialRatioGenerator();
$financialRatioGenerator->period = $period;
$financialRatioGenerator->SetDocumentInformation();

$financialRatioGenerator->AddPage();
$financialRatioGenerator->SetPageHeading();
$financialRatioGenerator->SetSustainabiltyAndProfitabilityRatios($period_id);
$financialRatioGenerator->ln();
$financialRatioGenerator->SetAssetAndLiabilityManagementRatios($period_id);
$financialRatioGenerator->ln();
$financialRatioGenerator->ln();
$financialRatioGenerator->SetPortfolioQualityRatios($period_id);
$financialRatioGenerator->ln();
$financialRatioGenerator->SetEfficiencyRatios($period_id);
try {
    $financialRatioGenerator->output();
    }
catch(Exception $e) {
    echo "Error: " . $e->getMessage();
}
//
?>
```