

Shree Swaminarayan College of Computer Science

(Affiliated to M. K. Bhavnagar University)

MASTER BUDGET

BY

Vinay Ashokbhai Dodiya
[25360053]

UNDER GUIDANCE OF

Mr. Nirav Shah

SUBMITTED TO

SHREE SWAMINARAYAN COLLEGE OF COMPUTER SCIENCE

FOR DEGREE OF

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY



Shree Swaminarayan College of Computer Science

(Affiliated to M. K. Bhavnagar University)

GURUKUL CAMPUS, SARDARNAGAR, BHAVNAGAR-364001

NAAC Accreditation Grade "B" Gurukul Campus, Sardarnagar,
Bhavnagar. 364001. (Guj)



COLLEGE CERTIFICATE

This is to certify that Mr./Ms. _____

Seat No. _____ student of Bachelor of Science in Information
Technology (B.Sc.IT) Semester-VI, in our institute has successfully completed
the project assigned to his/her at entitled _____

during the period December-2024 to April-2025 as a part of Bachelor of
Science in Information Technology (B.Sc.IT) syllabus.(2024-2025).

Date :

Place : Bhavnagar

Project Guide

I/C Principal

MASTER BUDGET

ACKNOWLEDGMENT

We are deeply grateful to everyone who directly or indirectly contributed to the development of this project.

First and foremost, we sincerely thank **Dr. Paresh Rathod**, the I/C Principal, and **Dr. Kalpesh Gundigara**, the Head of the Department at SSCCS, for providing us with a creative and encouraging environment for project development. Their constant support and openness to new ideas have greatly motivated us throughout this journey.

We would also like to extend our heartfelt gratitude to our guide and faculty, **Mr. Nirav Shah**, for giving us the golden opportunity to work on this exciting project "MASTER BUDGET". His guidance has helped us explore new concepts and enhance our research skills.

Lastly, we express our sincere appreciation to everyone who stood by us, offering support and assistance whenever we faced challenges.

This project is not just about academic achievement; it is a journey of learning and growth.

"THANK YOU TO EVERYONE WHO HELPED US!"

Submitted By:

Vinay Dodiya [25360053]

Shree Swaminarayan College of Computer Science

MASTER BUDGET

Index

INTRODUCTION	5
CHAPTER - 1 INTRODUCTION	6
BACKGROUND	6
OBJECTIVE	6
PURPOSE	7
SCOPE	7
APPLICABILITY	7
CHAPTER - 2 REQUIREMENT AND ANALYSIS	8
PROBLEM DEFINITION	8
REQUIREMENT SPECIFICATION	8
PLANNING AND SCHEDULING	9
DEVELOPER REQUIREMENT	10
CHAPTER - 3 SYSTEM DESIGN	11
SYSTEM ANALYSIS	11
Existing System	11
Limitations of the Existing System:	11
Proposed System	11
System Tools	12
DATA DICTIONARY	15
Data Flow Diagram	20
Entity Relationship Diagram (ERD)	29
DESIGNING	31
General webpage	31
User side webpage	35
Admin side webpage	43
CHAPTER- 4 TESTING AND IMPLEMENTATION	49
TESTING	49
Test Plan	50
Testing Strategies	50
Level of Testing	55
Types of Testing	58
Testing Methods	62
TEST CASE	65
CHAPTER- 5 CONCLUSION	74
LIMITATION OF SYSTEM	74
FUTURE SCOPE OF APPLICATION	74
BIBLIOGRAPHY	75

MASTER BUDGET

INTRODUCTION

Master Budget is an advanced web-based financial management tool designed to streamline budgeting, expense tracking, and financial analysis. It offers users a seamless platform to record transactions, categorize expenses, and visualize financial data through interactive charts and reports. Whether for personal finance, business operations, or professional financial planning, Master Budget enhances decision-making with real-time insights and intuitive budgeting features. Its user-friendly interface ensures easy navigation, while robust security measures safeguard financial information, making it a reliable solution for effective financial management.

MASTER BUDGET

Chapter - 1 Introduction

BACKGROUND

- The Master Budget project is a comprehensive initiative to develop a financial management platform designed for both individuals and organizations. It addresses the growing need for an efficient, secure, and accessible tool to manage income, expenses, and investments. By integrating modern technologies and user-focused features, the platform aims to simplify financial planning and enhance decision-making. The Master Budget project is a comprehensive initiative to develop a financial management platform designed for both individuals and organizations. It addresses the growing need for an efficient, secure, and accessible tool to manage income, expenses, and investments. By integrating modern technologies and user-focused features, the platform aims to simplify financial planning and enhance decision-making.

OBJECTIVE

- **Streamline Financial Management** – Provide a user-friendly platform for tracking income, expenses, and overall financial performance.
- **Enhance Budgeting Efficiency** – Enable users to create, manage, and adjust budgets dynamically to maintain financial stability.
- **Improve Financial Visibility** – Offer real-time insights through interactive dashboards and graphical representations for better decision-making.
- **Simplify Expense Tracking** – Categorize transactions automatically to help users monitor spending patterns and control costs.
- **Support Business and Personal Finance** – Cater to individuals, business owners, and financial professionals with tailored financial tools.
- **Ensure Data Security and Privacy** – Implement robust security measures to protect sensitive financial information.

MASTER BUDGET

PURPOSE

The purpose of this project is to:

- Empower users with tools to monitor their financial health.
- Simplify the complexities of budget management and tax planning.
- Offer a customizable platform tailored to diverse financial needs.

SCOPE

The Master Budget system will:

- Be accessible through a web-based interface.
- Support multiple currencies and customizable categories.
- Include secure login for administrators and end users.
- Provide exportable financial reports for a selected date range.

APPLICABILITY

The platform is applicable to:

- Individuals seeking personal financial management tools.
- Small and medium enterprises needing structured budget planning.
- Financial advisors and tax professionals looking for an integrated solution.

MASTER BUDGET

Chapter - 2 Requirement and Analysis

PROBLEM DEFINITION

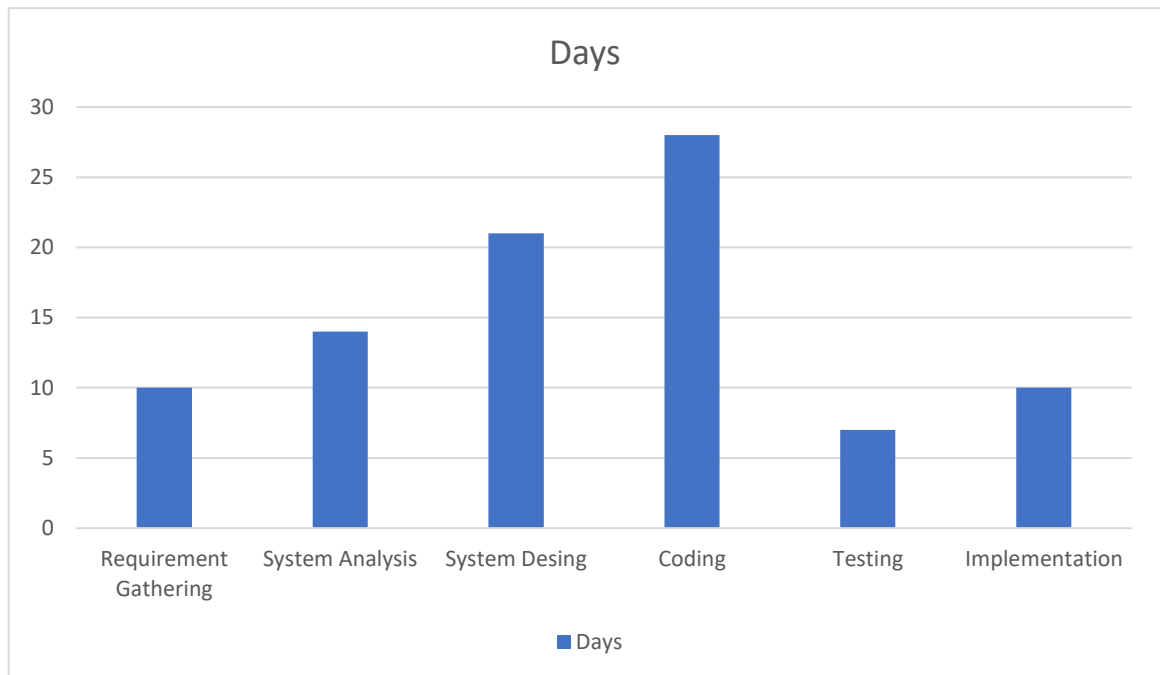
- A master budget is a comprehensive financial plan that consolidates all of an organization's budgets into a single document. It serves as a roadmap for the company's financial activities over a specific period, typically a fiscal year. The problem definition for a master budget website involves identifying the challenges organizations face in creating, managing, and analysing their master budgets. These challenges may include lack of integration between different departmental budgets, difficulties in forecasting revenues and expenses accurately, and the need for real-time data to make informed decisions. Additionally, the website must address the need for user-friendly tools that facilitate collaboration among various stakeholders, ensuring that all parts of the organization are aligned with the overall financial goals. By providing solutions to these issues, the master budget website can enhance the budgeting process, improve financial planning, and ultimately contribute to the organization's success.

REQUIREMENT SPECIFICATION

- **Functional:**
 - Secure login for administrators and end users.
 - Customizable categories for income, expenses, and investments.
 - Multi-currency support.
 - Reporting module with export options (PDF).
 - Investment suggestions for tax benefits.
- **Non-Functional:**
 - Non-functional requirements are mostly quality-related requirements which include the area of performance, availability, reliability, usability, flexibility, configurability, integration, maintainability, portability, and testability.

MASTER BUDGET

PLANNING AND SCHEDULING



MASTER BUDGET

DEVELOPER REQUIREMENT

- **Software Requirement:**
 - **Browser:** Chrome
 - **Operating System:** Compatible with all major Operating systems
 - **Frontend:** PHP 8.3
 - **Backend:** MySQL 8.0
 - **Software:** XAMPP 8.2.12
 - **Design:** HTML-5, JavaScript, CSS

- **Hardware Requirement:**
 - **Processor:** Compatible with all Intel Core processors
 - **RAM:** 2 GB or above
 - **Hard Disk:** 10 GB

MASTER BUDGET

Chapter - 3 System Design

SYSTEM ANALYSIS

- System analysis involves gathering and interpreting facts, diagnosing problems, and recommending improvements. It is a crucial phase in system development, requiring close communication between users and developers. The entire system is examined, inputs are identified, and potential issues are analysed. Proposed solutions are reviewed, modified as needed, and finalized based on user satisfaction.

Existing System

- In traditional financial management, users often rely on manual methods such as spreadsheets, paper records, or standalone accounting software to track their income and expenses. While these methods offer basic financial tracking.

Limitations of the Existing System:

Existing system come with several limitations, including:

- **Time-Consuming Processes** – Manual data entry and calculations increase the risk of errors and inefficiencies.
- **Limited Real-Time Insights** – Most existing systems do not provide real-time financial updates or interactive visualizations.
- **Lack of Integration** – Users need to manage multiple tools for budgeting, expense tracking, and reporting, leading to data fragmentation.
- **Poor Accessibility** – Traditional systems often lack cloud-based access, making it difficult to manage finances on the go.
- **Security Risks** – Many existing systems do not offer advanced encryption or secure data storage, increasing vulnerability to data breaches.

Proposed System

- These limitations highlight the need for a modern, web-based financial management tool like Master Budget, which provides automation, real-time tracking, intuitive dashboards, and enhanced security to streamline financial planning and decision-making.

MASTER BUDGET

System Tools

1.1) PHP:

- The term PHP is an acronym for PHP: Hypertext Pre-processor. PHP is a server-side scripting language designed specifically for web development.
- PHP can actually do anything related to server-side scripting or more popularly known as the backend of a website.
- For example, PHP can receive data from forms, generate dynamic page content, can work with databases, create sessions, send and receive cookies, send emails etc.
- There are also many hash functions available in PHP to encrypt user's data that makes PHP secure and reliable to be used as a server-side scripting language. So, these are some of the abilities of PHP that makes it suitable to be used as server-side scripting language.



1.2) MySQL:

- MySQL is an open-source relational database management system (RDBMS). It is the most popular database system used with PHP. MySQL is developed, distributed, and supported by Oracle Corporation.
- The data in a MySQL database are stored in tables which consists of columns and rows.
 - MySQL is a database system that runs on a server.
 - MySQL is ideal for both small and large applications.
 - MySQL is very fast, reliable and easy to use database system. It uses standard SQL
 - MySQL compiles on a number of platforms.



MASTER BUDGET

1.3) XAMPP:

- XAMPP is a cross-platform web server that is free and open-source.
- XAMPP is a short form for Cross-Platform, Apache, MySQL, PHP, and Perl.
- XAMPP is a popular cross-platform web server that allows programmers to write and test their code on a local webserver.
- It was created by Apache Friends, and the public can revise or modify its native source code.
- It includes MariaDB, Apache HTTP Server, and interpreters for PHP and Perl, among other computer languages.
- Because of XAMPP's simplicity of deployment, a developer can quickly and easily install a WAMP or LAMP stack on an operating system, with the added benefit that common add-in apps like WordPress and Joomla can also be loaded.



1.4) HTML:

- HTML for Hyper Text Markup Language. It is used to design web pages using markup language.
- HTML is the combination of Hypertext and Markup language.
- Hypertext defines the link between the web pages. Markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly.



MASTER BUDGET

- Most markup languages (e.g. HTML) are human readable. Language uses tags to define
- what manipulation has to be done on the text. HTML is a Markup language used by the browser to manipulate text, images and other content, in order to display it in the required format.
- HTML uses predefined tags and elements which tell the browser how to properly display the content. Remember to include closing tags. If omitted, the browser applies the effect of the opening tag until the end of page. Attribute tell us more about elements.
- Attribute provide additional information about the contents of an elements. They appear on the opening tag of the element and are made up of two parts: a name and a value, separated by an equal's sign.

1.5) Cascading Style System (CSS):

- Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page. CSS is easy to learn and understood but it provides powerful control over the presentation of an HTML document.



1.6) JAVASCRIPT:

- JavaScript is a very powerful client-side scripting language. JavaScript is used mainly for enhancing the interaction of a user with the webpage.
- In other words, you can make your webpage livelier and more interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development.



MASTER BUDGET

DATA DICTIONARY

- A data dictionary is a file that helps to define the organization of a particular database. The data dictionary acts as a description of the data objects or items in a model and is used for the benefit of the programmer or other people who may need to access it.
- A data dictionary does not contain the actual data from the database; it contains only information for how to describe/manage the data; this is called metadata*. Building a data dictionary provides the ability to know the kind of field, where it is located in a database, what it means, etc. It typically consists of a table with multiple columns that describe relationships as well as labels for data.
- **A data dictionary often contains the following information about fields:**
 - Default values
 - Constraint information
 - Definitions (example: functions, sequence, etc.)
 - The amount of space allocated for the object/field
 - Auditing information
- **What is the data dictionary used for?**
 - It can also be used as a read-only reference in order to obtain information about the database.
 - A data dictionary can be of use when developing programs that use a data model.
 - The data dictionary acts as a way to describe data in “real-world” terms.
- **Why is a data dictionary needed?**
 - One of the main reasons a data dictionary is necessary is to provide better accuracy, organization, and reliability in regards to data management and user/administrator understanding and training.

MASTER BUDGET

- **Benefits of using a data dictionary:**

1. Improved data quality
2. Consistency in data use
3. Improved documentation and control of data
4. Faster and easier data analysis
5. Easier Programming
6. Better trust in data integrity

1) Table: Accounts

Table Description: Table for user's details or login details.

Sr. no.	Field Name	Type	Size	Constraints	Description
1	Name	Varchar	20	Not Null	Name of User
2	Email	Varchar	30	Primary Key	Email ID of User
3	Country	Varchar	20	Not Null	Country of User (Reference Currency (Country_name))
4	Password	Varchar	20	Not Null	Password for Login

2) Table: Bank_details

Table Description: Table for add bank account.

Sr. no.	Field Name	Type	Size	Constraints	Description
1	bankName	Varchar	20	Not Null	Name of Bank
2	accountNo	Varchar	11	Not Null	Account Number
3	ifscCode	Varchar	11	Not Null	IFSC Code of Bank Branch
4	email	Varchar	30	Not Null	Email id of User (Reference Account (Email))
5	PhoneNo	Char	10	Not Null	Phone Number
6	branch	Varchar	20	Not Null	Name of Bank Branch
7	Pincode	Char	6	Not Null	Pincode of Branch
8	date	Datetime	-	-	Date

Shree Swaminarayan College of Computer Science

MASTER BUDGET

3) Table: Category

Table Description: Table for store categories.

Sr. no.	Field Name	Type	Size	Constraints	Description
1	category_ID	int	11	Primary Key	Category ID
2	accountNo	Varchar	11	Not Null	Account Number (Reference Bank_details (accountNo))
3	category_name	Varchar	20	Not Null	Name of Category
4	type	Varchar	10	Not Null	Category type (Income / Expense)
5	date	datetime	-	-	Date

4) Table: Currency

Table Description: Table for country and currency details.

Sr. no.	Field Name	Type	Size	Constraints	Description
1	country_id	Int	3	Primary Key	Country ID
2	Country_name	Varchar	20	Not Null	Country Name
3	currency	Varchar	30	Not Null	Name of Currency
4	Symbol	Varchar	10	Not Null	Symbol of Currency
5	crvalue	Double	-	-	Currency in US Dollar like (1 USD = 86.03 INR)

5) Table: Transactions

Table Description : Table for store all transactions.

Sr. no.	Field Name	Type	Size	Constraints	Description
1	account	Varchar	20	Not Null	Account Number (Reference Account (accountNo))
2	tran_id	Double	-	Primary Key	Transaction ID
3	trannname	Varchar	20	Not Null	Transaction Name
4	category	Varchar	20	Not Null	Category of Transaction

Shree Swaminarayan College of Computer Science

MASTER BUDGET

					(Reference Category (category_name))
5	tran_date	date	-	-	Transaction date
6	type	Varchar	20	Not Null	Transaction Type (Income / Expense)
7	method	Varchar	20	Not Null	Payment Method
8	amount	double	-	-	Transaction Amount
9	memo	Text	-	-	Description of Transaction
10	date	datetime	-	-	Current time when transaction perform

6) Table: Income

Table Description : Table for store only income transactions.

Sr. no.	Field Name	Type	Size	Constraints	Description
1	account	Varchar	20	Not Null	Account Number (Reference Account (accountNo))
2	tran_id	Double	-	Primary Key	Transaction ID
3	trannname	Varchar	20	Not Null	Transaction Name
4	category	Varchar	20	Not Null	Category of Transaction (Reference Category (category_name))
5	type	Varchar	20	Not Null	Transaction Type (Income / Expense)
6	method	Varchar	20	Not Null	Payment Method
7	tran_date	date	-	-	Transaction date
8	amount	double	-	-	Transaction Amount
9	balance	double	-	-	Balance

7) Table: Expense

Table Description : Table for store only expense transactions.

Shree Swaminarayan College of Computer Science

MASTER BUDGET

Sr. no.	Field Name	Type	Size	Constraints	Description
1	account	Varchar	20	Not Null	Account Number (Reference Account (accountNo))
2	tran_id	Double	-	Primary Key	Transaction ID
3	tranname	Varchar	20	Not Null	Transaction Name
4	category	Varchar	20	Not Null	Category of Transaction (Reference Category (category_name))
5	type	Varchar	20	Not Null	Transaction Type (Income / Expense)
6	method	Varchar	20	Not Null	Payment Method
7	tran_date	date	-	-	Transaction date
8	amount	double	-	-	Transaction Amount
9	balance	double	-	-	Balance

8) Table: Investments

Table Description: Table for Investment details.

Sr. no.	Field Name	Type	Size	Constraints	Description
1	ID	int	3	Primary Key	Investment ID
2	Item	Varchar	100	Not Null	Investment Name
3	Text	text	-	Not Null	Description of investments

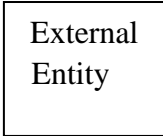
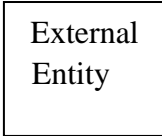
MASTER BUDGET

Data Flow Diagram

- A **Data Flow Diagram (DFD)** is a visual representation of how data moves through a system. It is used to model the flow of data between processes, data stores, and external entities. DFDs are hierarchical, meaning they can be broken down into levels of detail, starting from a high-level overview (Context Diagram) to more detailed levels (Level 1, Level 2, etc.).

Components of DFD

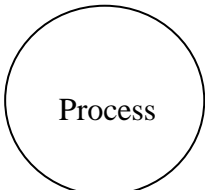
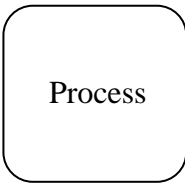
- **External Entity**

Symbols	Author	
	Yourdon & Peter Coad	Gane & Sarson
External Entity		

Description:

- This External Entity is use for write a user name, customer name, etc.

- **Process**



Symbols	Author	
	Yourdon & Peter Coad	Gane & Sarson
Process		

Description:

- This Process is use for write a Validation, Verification, etc.

MASTER BUDGET



- **Database**

Symbols	Author	
	Yourdon & Peter Coad	Gane & Sarson
Database		

Description:

- The Database should be / must be 1 side or 2 side open otherwise that are not called database.
- Database also called Data Store.

- **Flow of Data**

Symbols	Author	
	Yourdon & Peter Coad	Gane & Sarson
Flow of Process		

Description:

- This is a flow of process symbol; it uses to describe the flow of process that where to start & where to end.
- The flow of process symbol also known by Structure Pointer & Data Flow.

MASTER BUDGET

- 0 Level DFD for Master Budget System

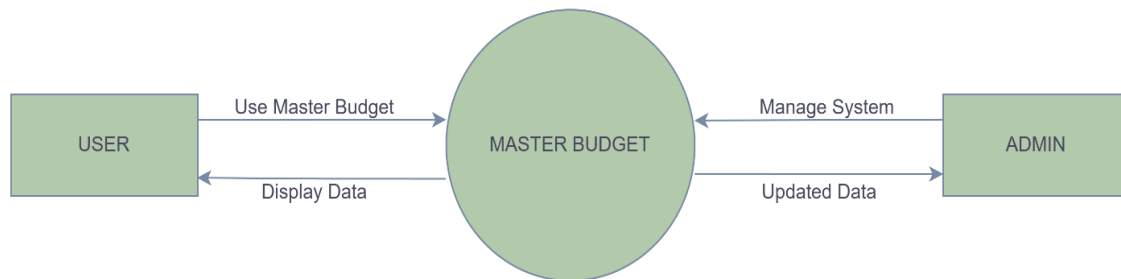


Figure 1 context level DFD

MASTER BUDGET

- 1st Level DFD for User

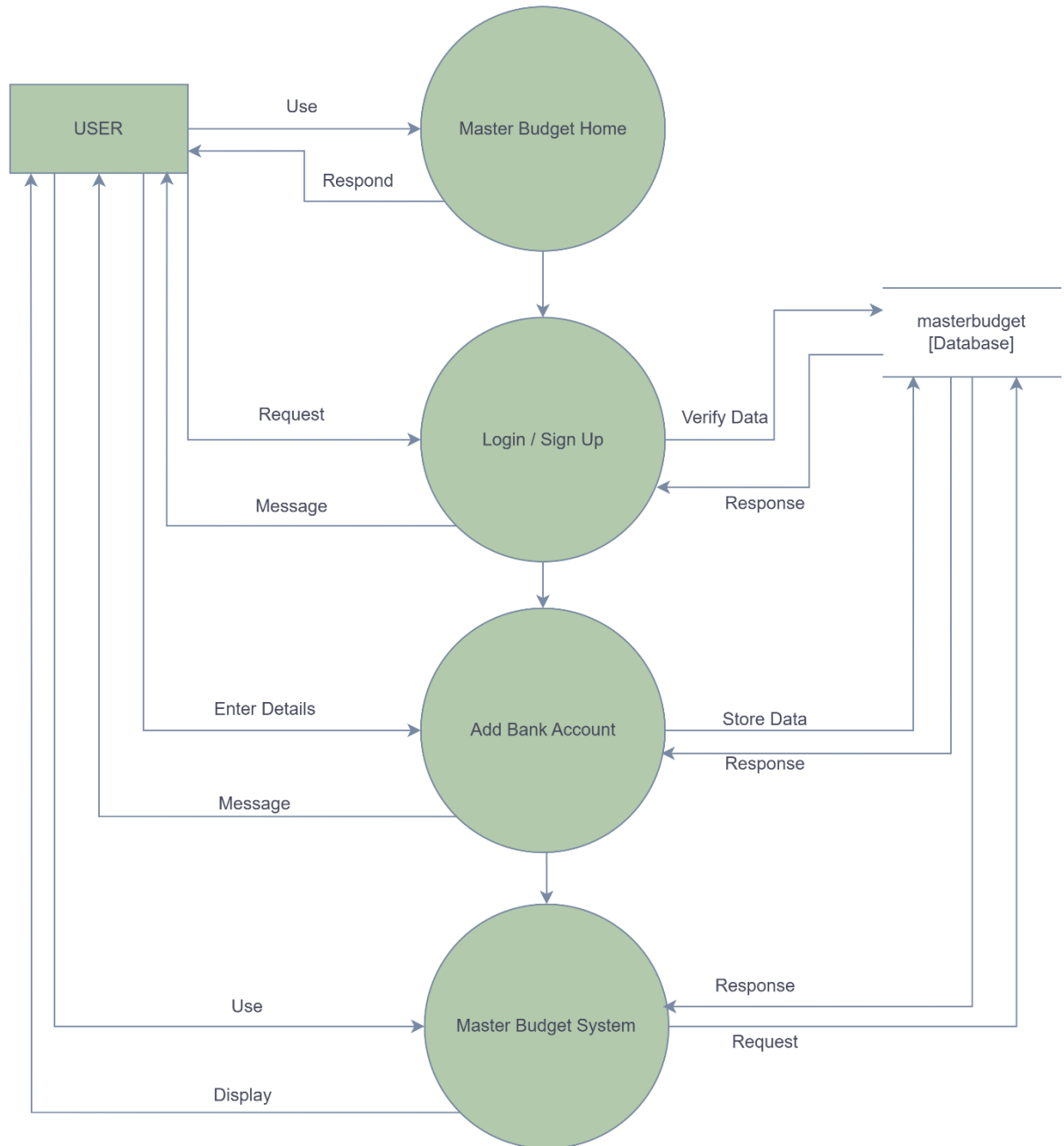


Figure 2 1st Level DFD for User

MASTER BUDGET

- 2^{dn} Level DFD for User

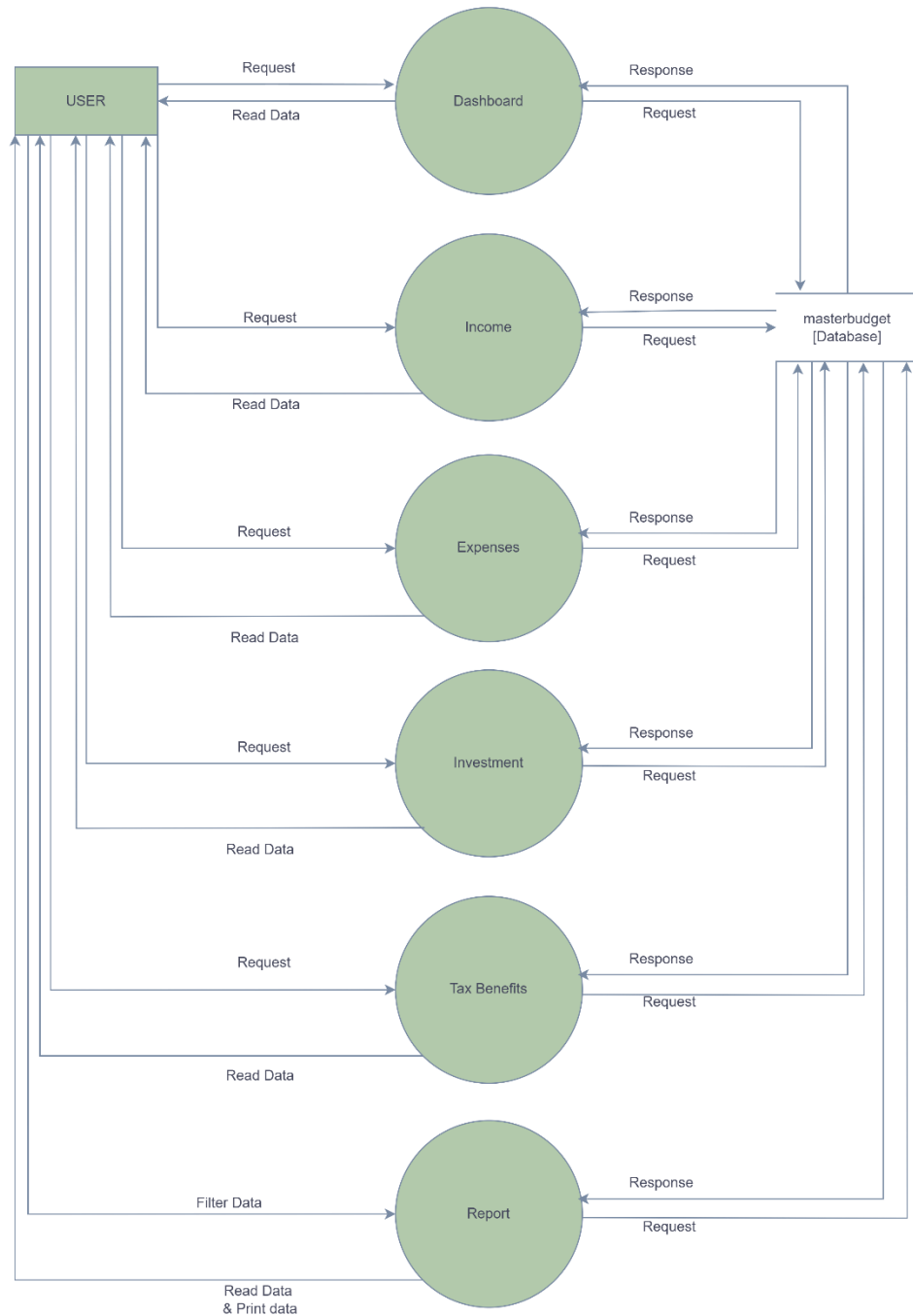


Figure 3 2^{dn} Level DFD for User

MASTER BUDGET

- 3rd Level DFD for User in Income

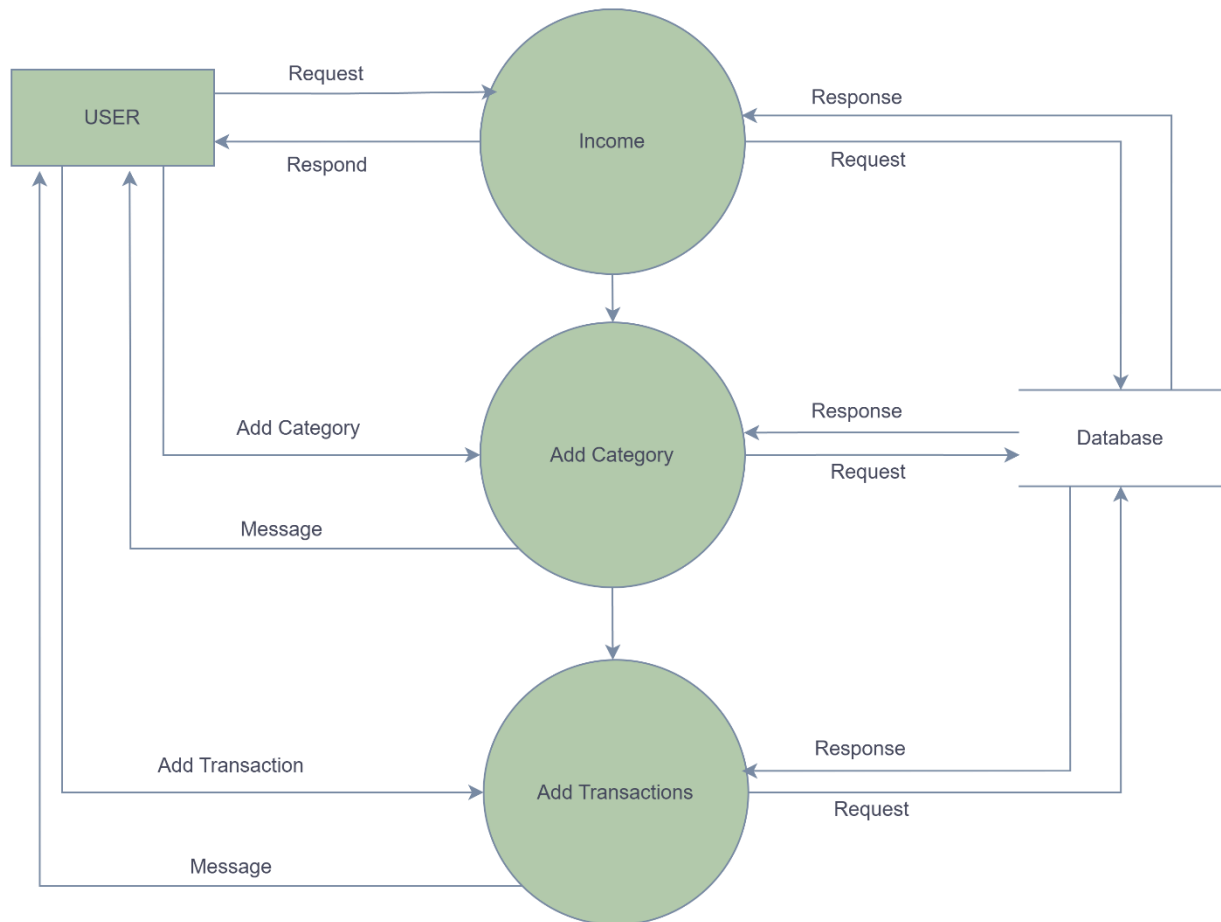


Figure 4 3rd Level DFD for User in Income

MASTER BUDGET

- 3rd Level DFD for User in Expenses

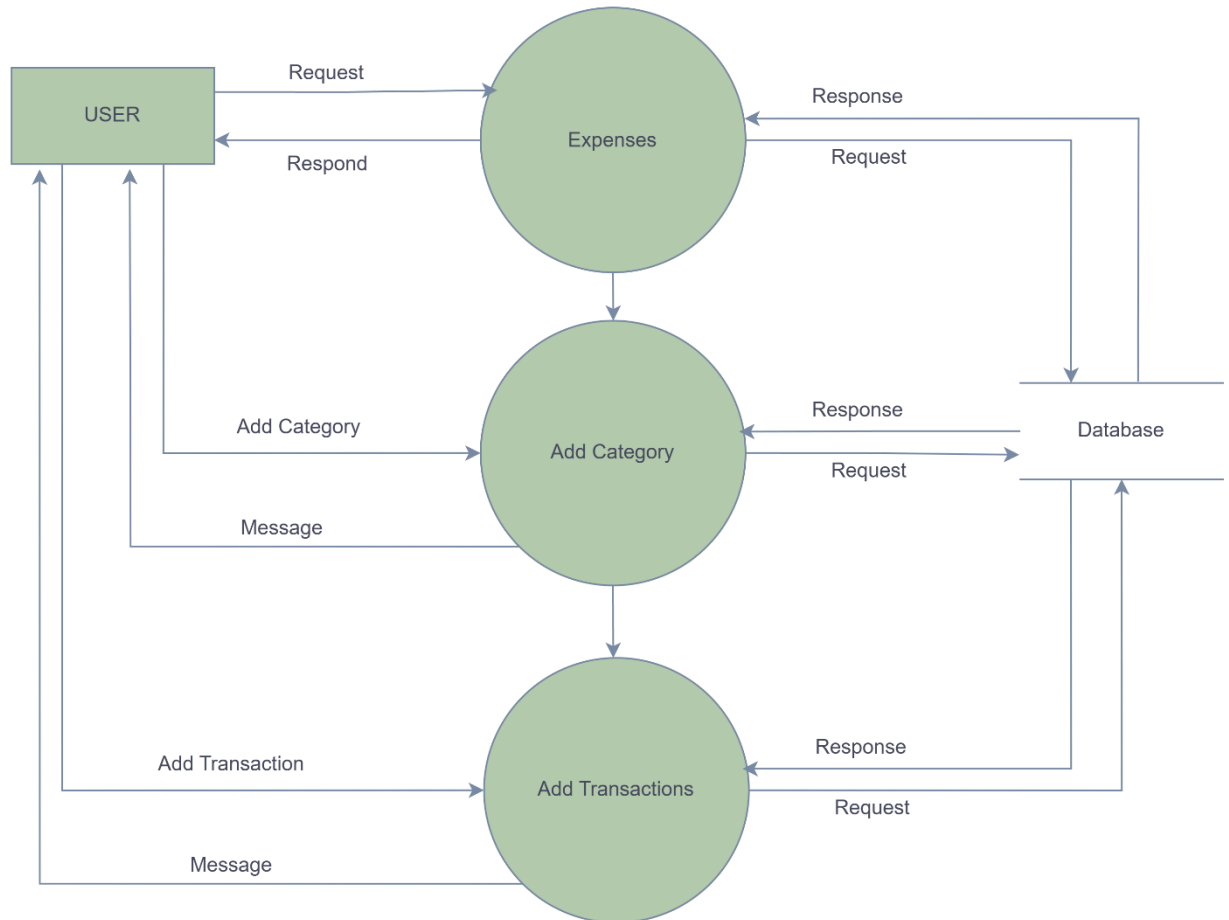


Figure 5 3rd Level DFD for User in Expenses

MASTER BUDGET

- 1st Level DFD for ADMIN

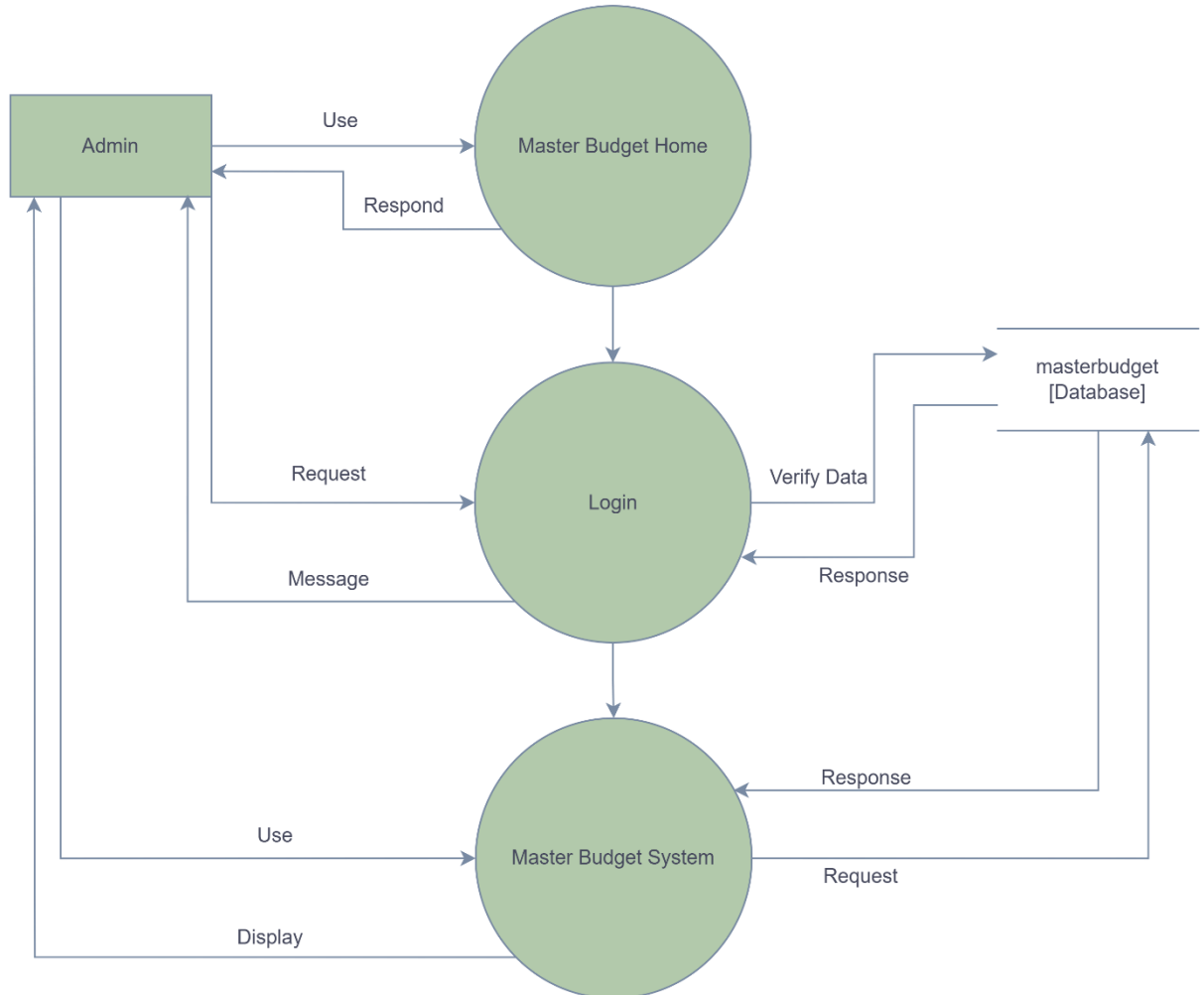


Figure 6 1st Level DFD for ADMIN

MASTER BUDGET

- 2nd Level DFD for ADMIN

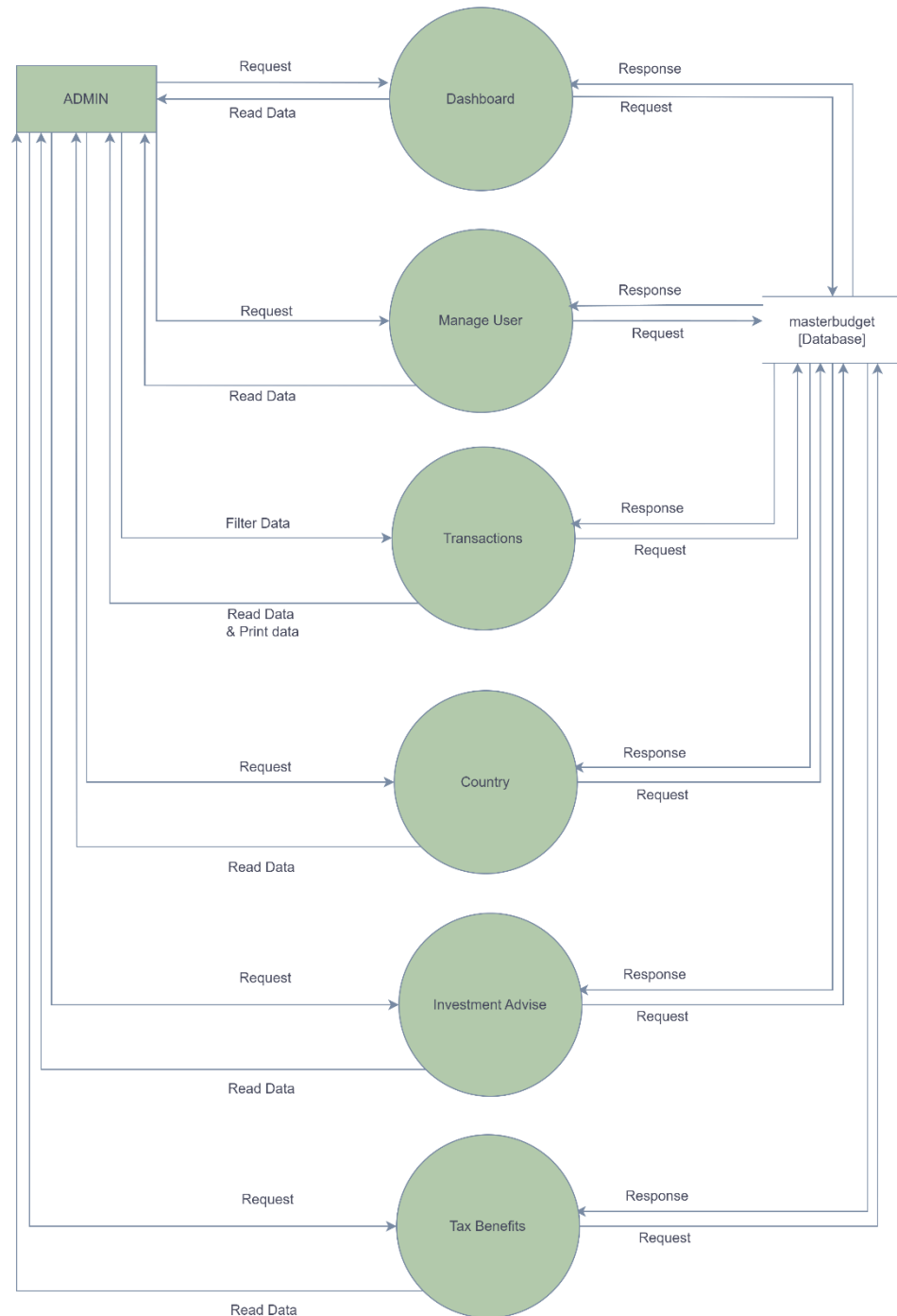


Figure 7 2nd Level DFD for ADMIN

Entity Relationship Diagram (ERD)

An entity relationship diagram is a visual representation of different data using conventions that describe how these data are related to each other. While able to describe just about any system, ER-diagram are most often associated in engineering software's and if networks.

ER-diagram are frequently used the design stage of a development process in order to identify different system element and their relationship with each other.

Types of ER Diagram

ER diagrams will differ on how they express cardinality. They will also differ in how they display entities and their attributes and whether or not they show relationships or attributes as separate symbols.

1. Traditional ERD
2. IDEF1X Notation ERD - Relational Schema

MASTER BUDGET

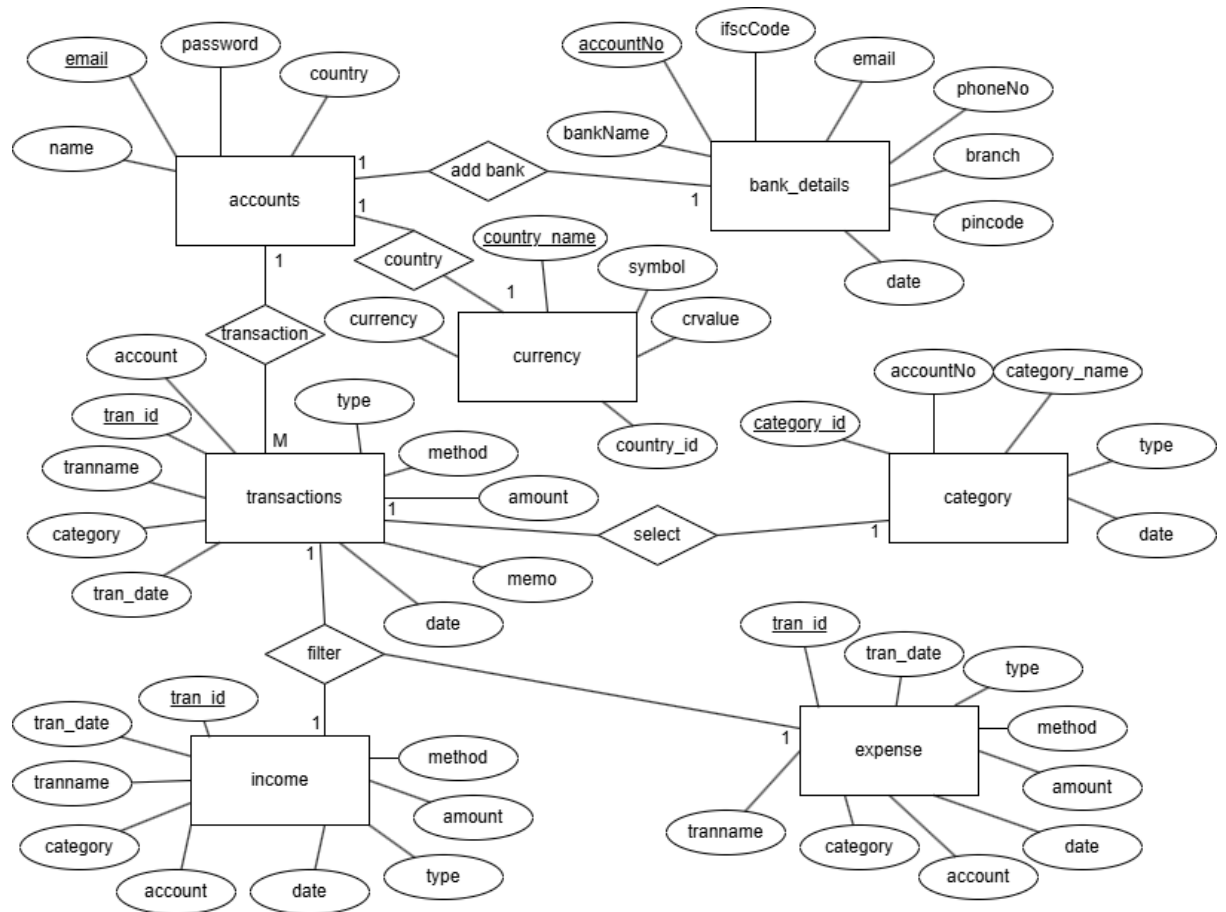


Figure 8 ER Diagram

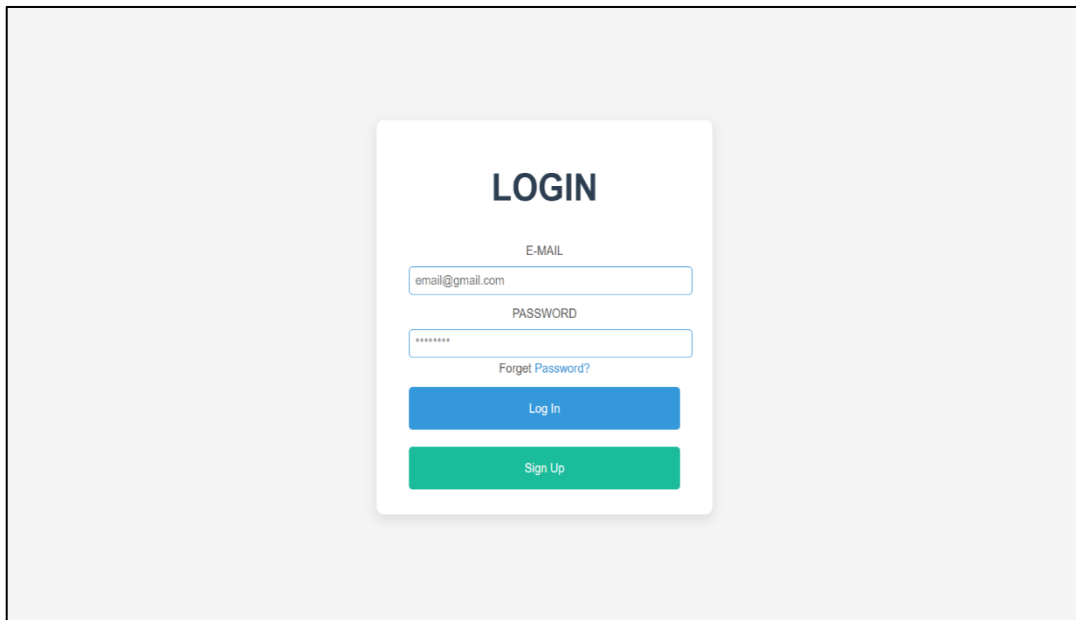
MASTER BUDGET

DESIGNING

General webpage

Login Page

- This is login page of both admin and agent.
- If username and password belong to admin then it will redirect admin panel otherwise it will redirect agent panel based on their role.



MASTER BUDGET

Index Page

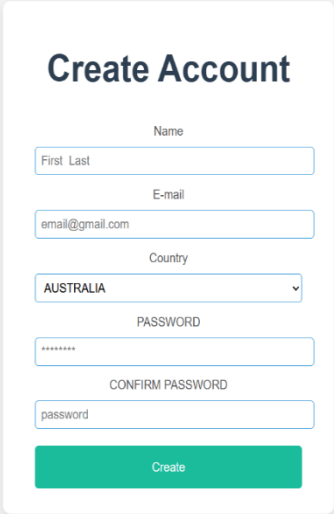
- This is index page for MASTER BUDGET Website.
It displays basic information of website.

The screenshot displays the index page of the MASTER BUDGET website. The page has a dark blue header with a logo on the left and navigation links (About, Features, Contact, Login / Sign Up) on the right. The main content area is white and features a large blue banner with the text 'MASTER BUDGET' and 'BUDGET YOUR PERSONAL FINANCE', followed by a green 'Get Start' button. Below this is a 'Features' section with six white cards: Dashboard, Income, Expense, Investment, Tax Benefits, and Reports, each with a blue arrow icon. The 'Contact US' section follows, with a form containing fields for Email, Subject, and Message, and a blue 'Submit' button. The footer is dark blue and includes social media links (LinkedIn, Twitter, Email), the text 'Master Budget - Your trusted finance manager.', and links for 'About Us' and 'Contact'. A copyright notice '© 2025 Master Budget. All Rights Reserved.' is at the bottom.

MASTER BUDGET

Sign Up Page

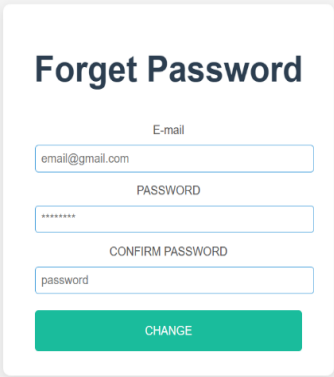
- This is Sign Up page for User.
- It is user for create account in website.



The image shows a 'Create Account' form centered on a light gray background. The form is white with a thin gray border. It contains the following fields: 'Name' (with a placeholder 'First Last'), 'E-mail' (with a placeholder 'email@gmail.com'), 'Country' (a dropdown menu showing 'AUSTRALIA'), 'PASSWORD' (with a placeholder '*****'), and 'CONFIRM PASSWORD' (with a placeholder 'password'). A green 'Create' button is at the bottom.

Forget Password Page

- This is Forget Password page for User.
- It is user for change password if user forget password.

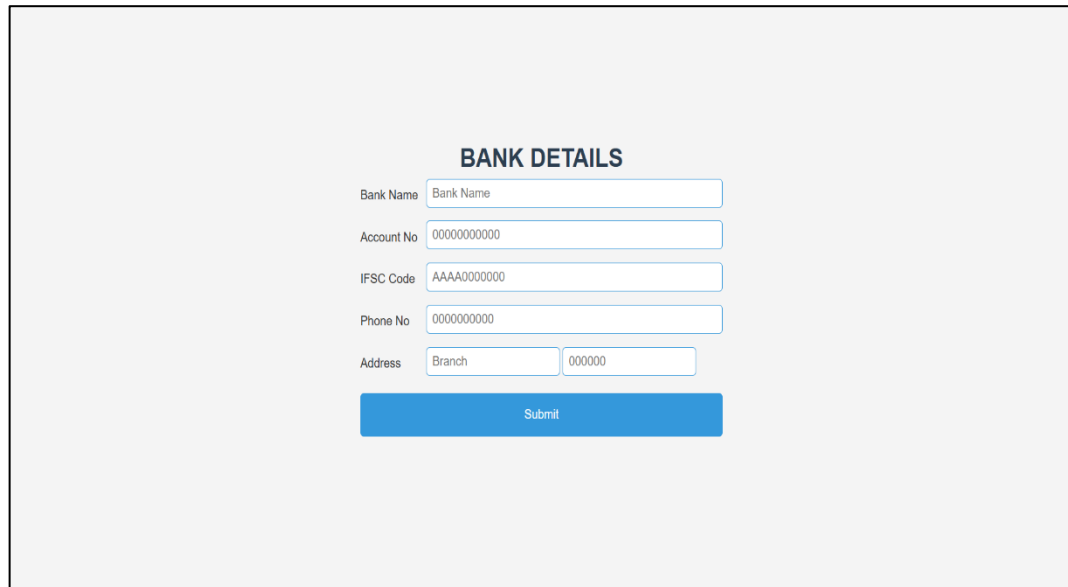


The image shows a 'Forget Password' form centered on a light gray background. The form is white with a thin gray border. It contains the following fields: 'E-mail' (with a placeholder 'email@gmail.com'), 'PASSWORD' (with a placeholder '*****'), and 'CONFIRM PASSWORD' (with a placeholder 'password'). A green 'CHANGE' button is at the bottom.

MASTER BUDGET

Add Bank Page

- This Add Bank details page for User.
- It allows enter bank details to user.



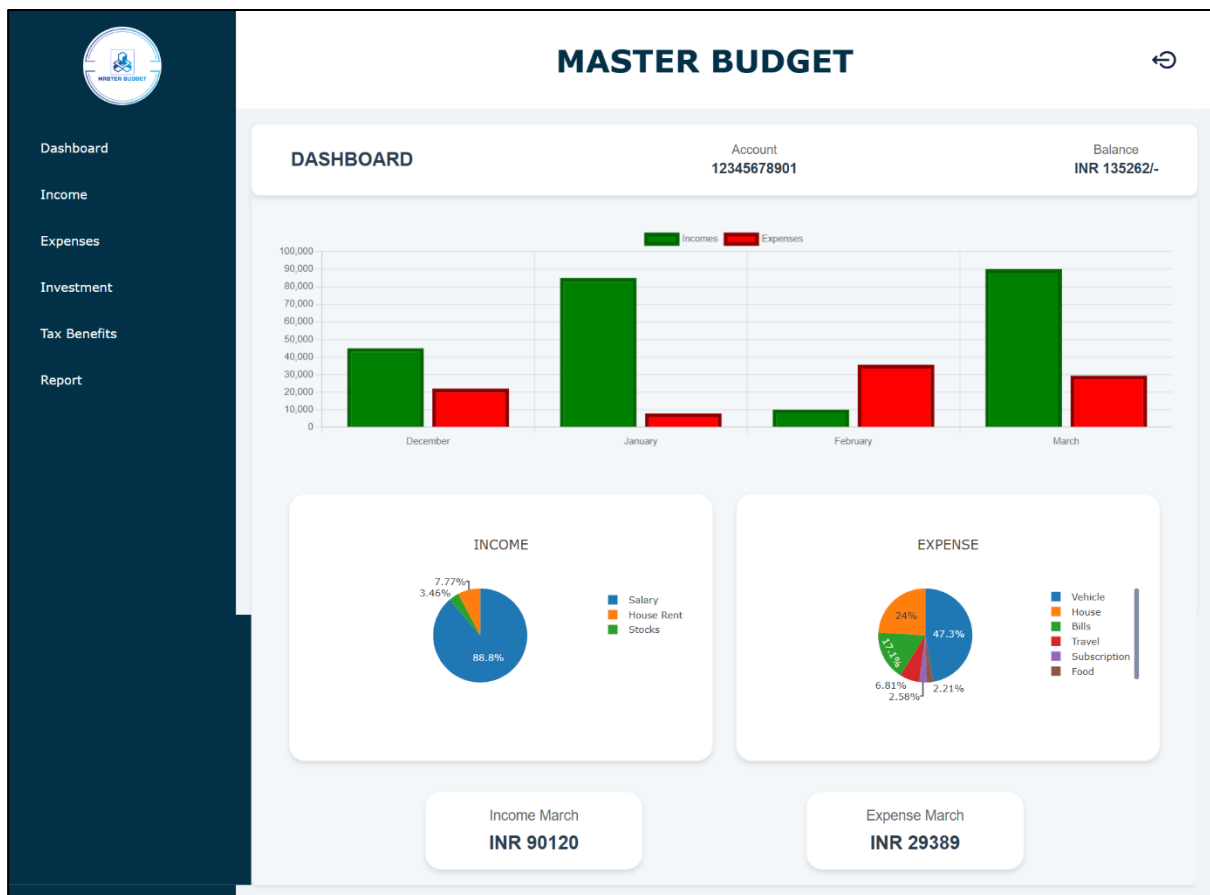
The screenshot displays a web form titled "BANK DETAILS" centered on a light gray background. The form contains the following fields: "Bank Name" (text input), "Account No" (text input with placeholder "0000000000"), "IFSC Code" (text input with placeholder "AAAA000000"), "Phone No" (text input with placeholder "0000000000"), and "Address" (split into two text inputs: "Branch" and "000000"). A blue "Submit" button is positioned at the bottom of the form.

MASTER BUDGET

User side webpage

Dashboard Page

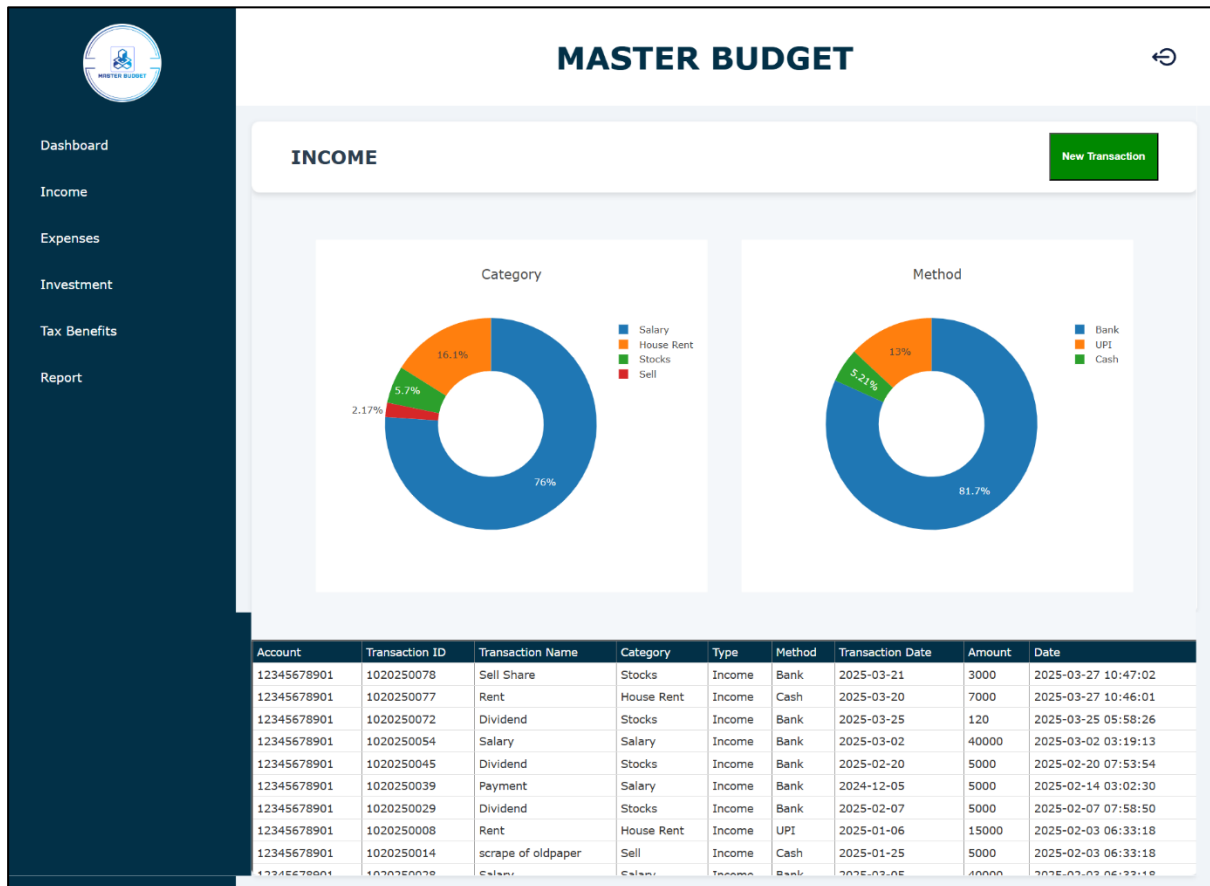
- This is Dashboard page for User.
- It displays user's transaction details in graphical format.
- Graph displays total incomes / expenses.
- Charts displays total incomes / expenses within a month.



MASTER BUDGET

Income Page

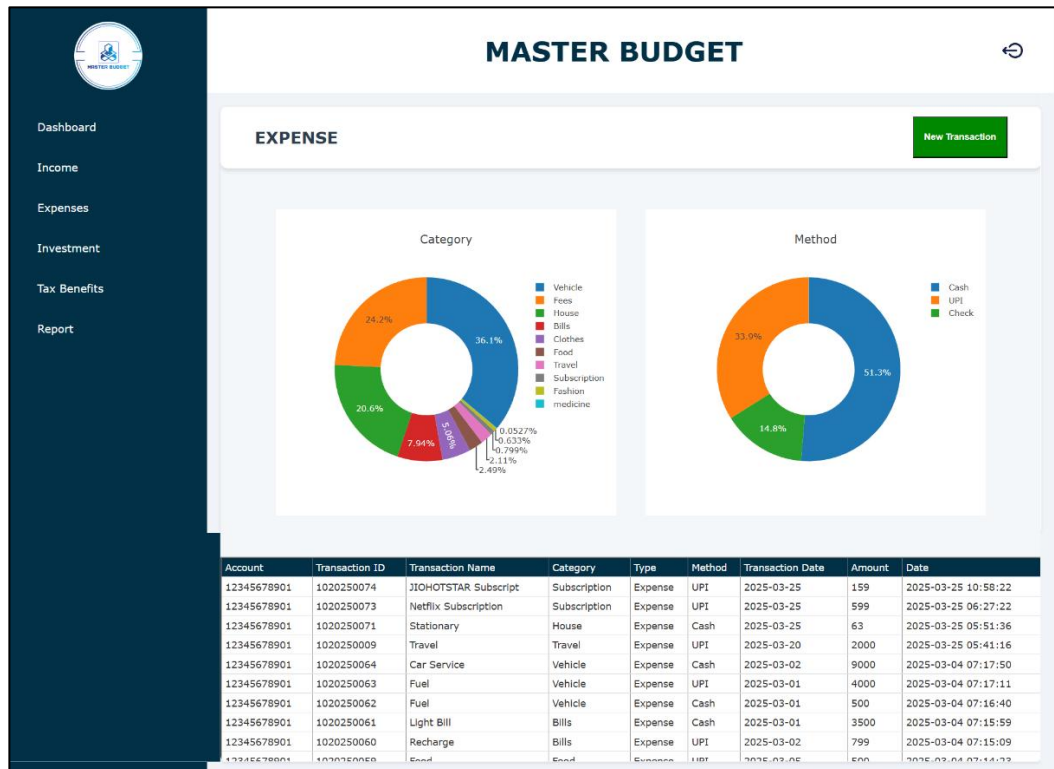
- This is Income page for User.
- Charts displays incomes by category / method.
- It displays total income transactions performed by user.
- It also uses for add new transactions.



MASTER BUDGET

Expense Page

- This is Expense page for User.
- Charts displays expense by category / method.
- It displays total expense transactions performed by user.
- It also uses for add new transactions.



MASTER BUDGET

New Transaction Page

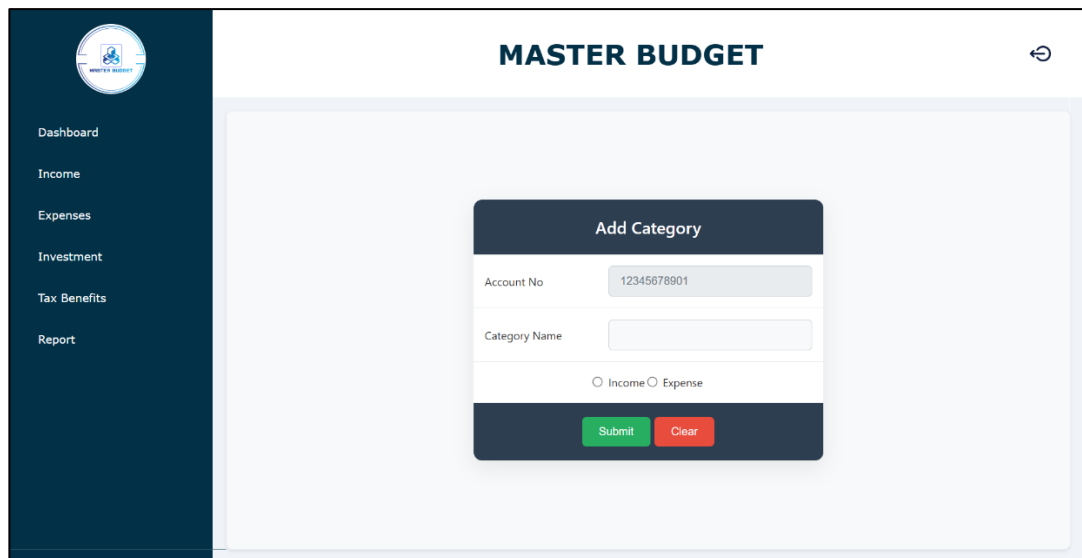
- This is New Transaction page for User.
- It allows to user for enter Transaction or Add New Category.

The screenshot displays the 'New Transaction' page of the 'MASTER BUDGET' application. On the left, a dark blue sidebar contains a logo and a list of navigation options: Dashboard, Income, Expenses, Investment, Tax Benefits, and Report. The main content area has a white background with a dark blue header bar that reads 'MASTER BUDGET'. Below this, a 'Transaction Details' form is centered. The form has a dark blue header with the title 'Transaction Details'. It contains several input fields: 'Transaction Name' (text input), 'Category' (dropdown menu showing 'House Rent' with a plus icon to add more), 'Date' (calendar icon and 'dd-mm-yyyy' format), a radio button to select between 'Income' and 'Expense', 'Method' (dropdown menu showing 'Cash'), 'Amount' (text input), and 'Memo' (text area). At the bottom of the form are two buttons: 'Save' (green) and 'Reset' (red).

MASTER BUDGET

Add Category Page

- This is Add Category page for User.
- User can add new category using it.
- Account no. automatically inserts in read-only mode.



The screenshot displays the 'Add Category' page of the 'MASTER BUDGET' application. On the left is a dark blue sidebar with a logo at the top and a list of navigation items: Dashboard, Income, Expenses, Investment, Tax Benefits, and Report. The main content area has a light gray background with the title 'MASTER BUDGET' at the top right. Centered on the page is a white modal form titled 'Add Category'. This form contains an 'Account No' field with the value '12345678901', a 'Category Name' input field, and radio buttons for 'Income' and 'Expense'. At the bottom of the form are two buttons: a green 'Submit' button and a red 'Clear' button.

MASTER BUDGET

Investment Page

- This is Investment page for User.
- It displays investment suggestions to user from database.

MASTER BUDGET

INVESTMENT

- Fixed Deposits**
A fixed deposit (FD) is a tenured deposit account provided by banks or non-bank financial institutions which provides investors a higher rate of interest than a regular savings account, until the given maturity date. It may or may not require the creation of a separate account. The term fixed deposit is most commonly used in India and the United States.
- Mutual Fund**
A mutual fund is a company that pools money from many investors and invests the money in securities such as stocks, bonds, and short-term debt. The combined holdings of the mutual fund are known as its portfolio. Investors buy shares in mutual funds. Each share represents an investor's part ownership in the fund and the income it generates.
- Debt Funds**
Debt funds invest in securities that generate fixed income, like treasury bills, corporate bonds, commercial papers, government securities, and many other money market instruments. All these instruments have a pre-decided maturity date and interest rate that the buyer can earn on maturity - hence the name fixed-income securities. The returns are usually not affected by fluctuations in the market. Therefore, debt securities are considered to be low-risk investment options.
- Post Office Time Deposit**
The post-office term deposit (POTD) is similar to a bank fixed deposit, where you save money for a definite time period, earning a guaranteed return through the tenure of the deposit. At the end of the deposit's tenure, the maturity amount comprises the capital deposited and the interest it earns.
- SIP**
A "SIP" (Systematic Investment Plan) means regularly investing a fixed amount of money into a mutual fund each month, essentially allowing you to buy more units when the market is low and fewer when it's high, effectively averaging out your purchase price and mitigating market volatility; for example, if you set up a ₹5,000 SIP in a mutual fund with a current NAV (Net Asset Value) of ₹50, you'd buy 100 units this month, but if the NAV drops to ₹40 next month, you'd buy 125 units with the same ₹5,000, lowering your overall average cost per unit.

MASTER BUDGET

Tax Benefits Page


- This is tax benefit page for User.
- It will calculate total taxable income and suggest some income invest for save tax.

The screenshot displays the 'MASTER BUDGET' application interface. On the left is a dark blue sidebar with a logo at the top and a list of menu items: Dashboard, Income, Expenses, Investment, Tax Benefits (highlighted), and Report. The main content area has a white header with the title 'MASTER BUDGET' and a refresh icon. Below the header is a light gray box containing a white card titled 'Tax Calculator and Savings Suggestions'. The card prompts the user to 'Enter Your Taxable Income (INR) :', with the value '112262' entered in the input field. A green button labeled 'Calculate Tax Benefits' is positioned below the input field. The results section, titled 'Tax Benefits and Suggestions', shows 'Taxable Income: (INR) 112,262' and 'Tax Payable: (INR) 0'. It also includes a 'Tax Saving Suggestions' section with a bullet point: 'Utilize the full limit of Section 80C (1.5 lakh) for tax savings.' A note at the bottom states: 'NOTE : (This Tax System is use according to United State Tax Policy)'.

MASTER BUDGET

Report Page

- This is Report page for User.
- It will display filtered transactions.
- Apply filter on transactions and display filtered data.
- Also download filtered data.



- Dashboard
- Income
- Expenses
- Investment
- Tax Benefits
- Report

MASTER BUDGET

Transactions

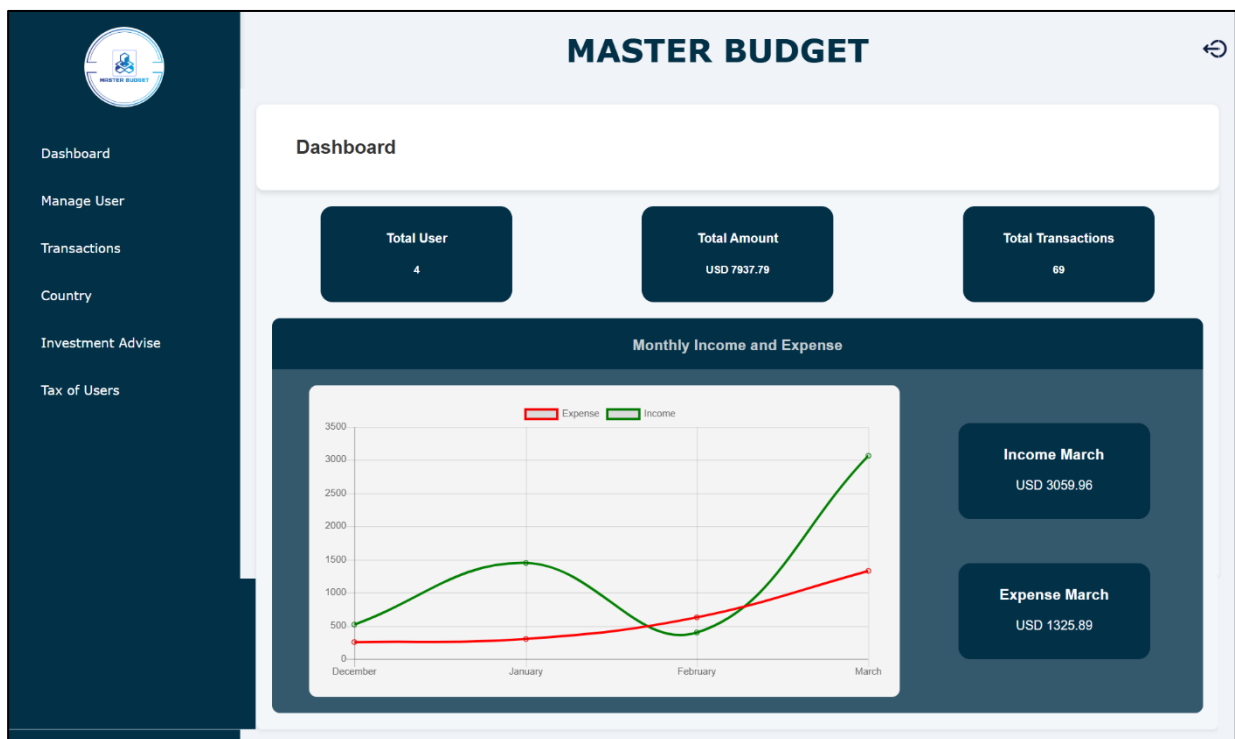
Transaction ID	Transaction Name	Type	Category	Method	Memo	Transaction Date	Date	Amount
1020250078	Sell Share	Income	Stocks	Bank	sell shares	2025-03-21	2025-03-27 10:47:02	3000
1020250077	Rent	Income	House Rent	Cash	House Rent	2025-03-20	2025-03-27 10:46:01	7000
1020250074	JIOHOTSTAR Subscript	Expense	Subscription	UPI	JIOHOTSTAR Subscription	2025-03-25	2025-03-25 10:58:22	159
1020250073	Netflix Subscription	Expense	Subscription	UPI	Netflix Recharge	2025-03-25	2025-03-25 06:27:22	599
1020250072	Dividend	Income	Stocks	Bank		2025-03-25	2025-03-25 05:58:26	120

MASTER BUDGET

Admin side webpage

Dashboard Page

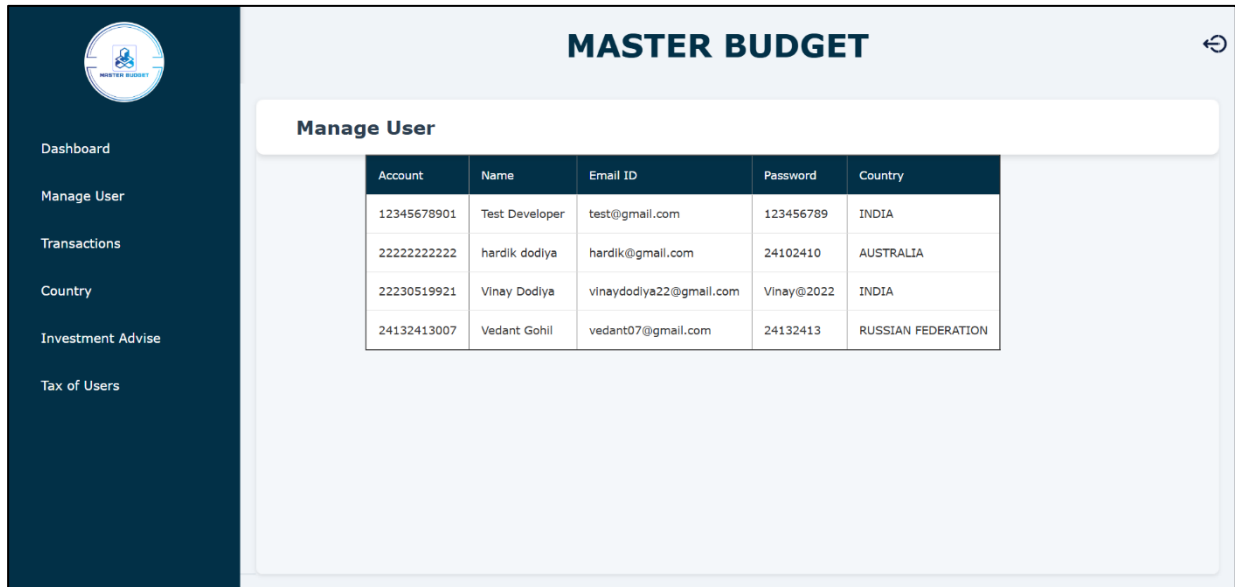
- This is Dashboard page for admin.
- It displays user's transaction details in graphical format.
- Graph displays total incomes / expenses.



MASTER BUDGET

Manage User Page

- This is Manage User page for admin.
- Display all user's information.



Account	Name	Email ID	Password	Country
12345678901	Test Developer	test@gmail.com	123456789	INDIA
22222222222	hardik dodiya	hardik@gmail.com	24102410	AUSTRALIA
22230519921	Vinay Dodiya	vinaydodiya22@gmail.com	Vinay@2022	INDIA
24132413007	Vedant Gohil	vedant07@gmail.com	24132413	RUSSIAN FEDERATION

MASTER BUDGET

Country Page

- This is Country page for Admin.
- Display all country from database and add country in database.



- Dashboard
- Manage User
- Transactions
- Country
- Investment Advise
- Tax of Users

MASTER BUDGET

Country

Country ID	Country Name	Currency	Symbol	Value
1	INDIA	INDIAN RUPEE	INR	86.5511
2	UNITED STATES	UNITED STATES DOLLAR	USD	1
3	EUROPEAN UNION	EURO	EUR	0.961
4	FRANCE	EURO	EUR	0.906

MASTER BUDGET

Investment Page

- This is Investment page for Admin.
- Enter investment suggestions that stores in database.
- Also displays investment suggestions from database.

MASTER BUDGET

Investment

[Add Investment](#) [View Investment](#)

Name:

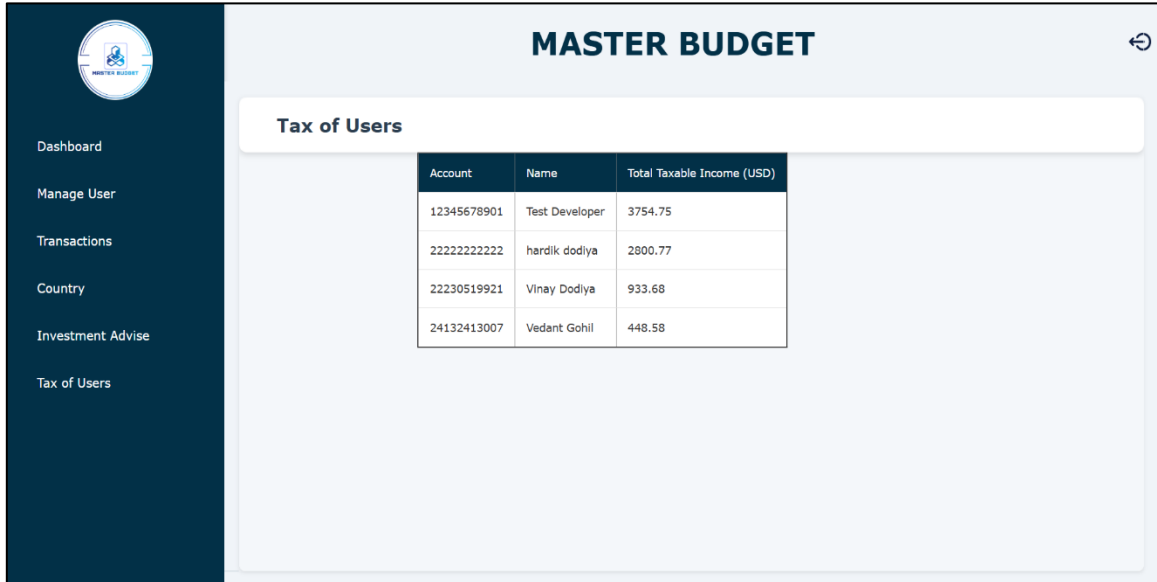
Description:

[Add](#)

MASTER BUDGET

Tax Benefits Page

- This is tax benefit page for Admin.
- It displays total taxable income of all user.




The screenshot shows a web application interface for the 'MASTER BUDGET' system. On the left is a dark blue sidebar with a circular logo at the top and a list of menu items: Dashboard, Manage User, Transactions, Country, Investment Advise, and Tax of Users. The main content area has a light blue header with the title 'MASTER BUDGET' and a refresh icon. Below the header, a white box titled 'Tax of Users' contains a table with three columns: Account, Name, and Total Taxable Income (USD). The table lists four users with their respective account numbers and taxable income values.

Account	Name	Total Taxable Income (USD)
12345678901	Test Developer	3754.75
22222222222	hardik dodiya	2800.77
22230519921	Vinay Dodiya	933.68
24132413007	Vedant Gohil	448.58

MASTER BUDGET

Transaction Page

- This is Report page for Admin.
- It will display filtered transactions of all user.
- Apply filter on transactions and display filtered data.
- Also download filtered data.



- Dashboard
- Manage User
- Transactions
- Country
- Investment Advise
- Tax of Users

MASTER BUDGET

Transactions

Filter

Transaction ID	Account	Transaction Name	Type	Category	Method	Memo	Transaction Date	Date	Amount
1020250089	22230519921	Myntra (T-Shirt Purc	Expense	Shopping	UPI	Myntra (T-Shirt Purchase)	2025-03-20	2025-03-27 11:37:52	899
1020250088	22230519921	Apollo Pharmacy (Med	Expense	Healthcare	Cash	Apollo Pharmacy (Medicines)	2025-03-14	2025-03-27 11:36:07	620
1020250087	22230519921	Ola Cabs	Expense	Transport	Cash	Ola Cabs (Airport Ride)	2025-03-18	2025-03-27 11:34:51	1196
1020250086	22230519921	Blinkit Groceries	Expense	House		Blinkit Groceries	2025-03-08	2025-03-27 11:33:05	1850

Download PDF

Chapter- 4 Testing and Implementation

TESTING

Software Testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.

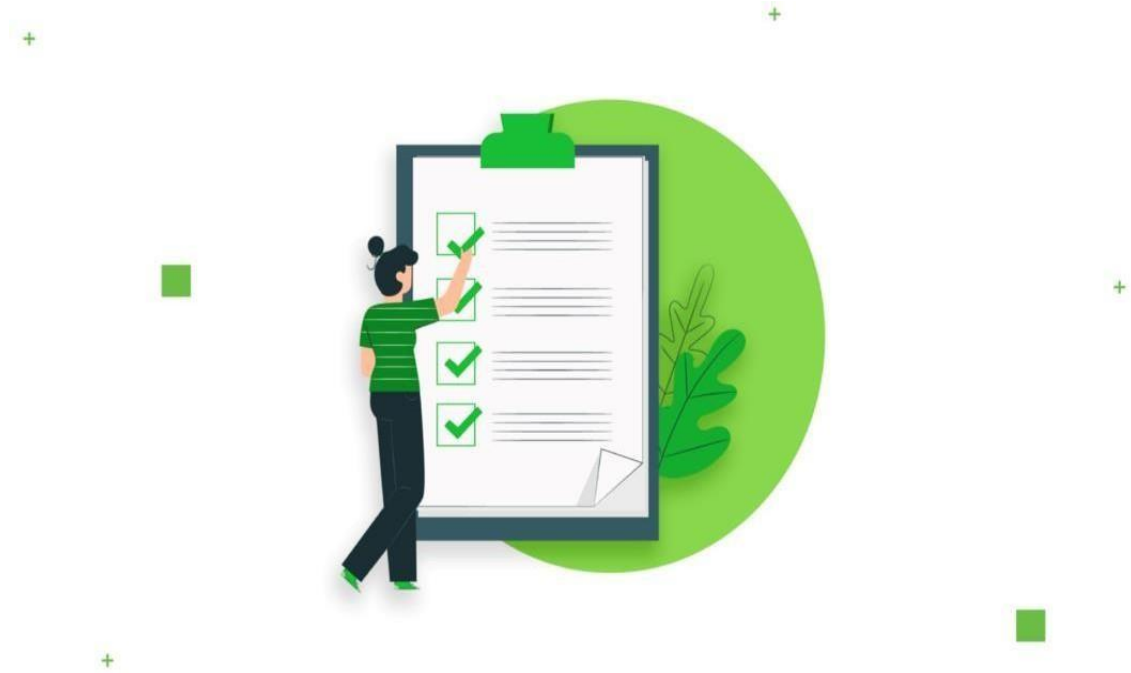


Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a White box and Black Box Testing.

MASTER BUDGET

Test Plan

A test plan is a detailed document that describes the test strategy, objectives, schedule, estimation and deliverables and resources required for testing.

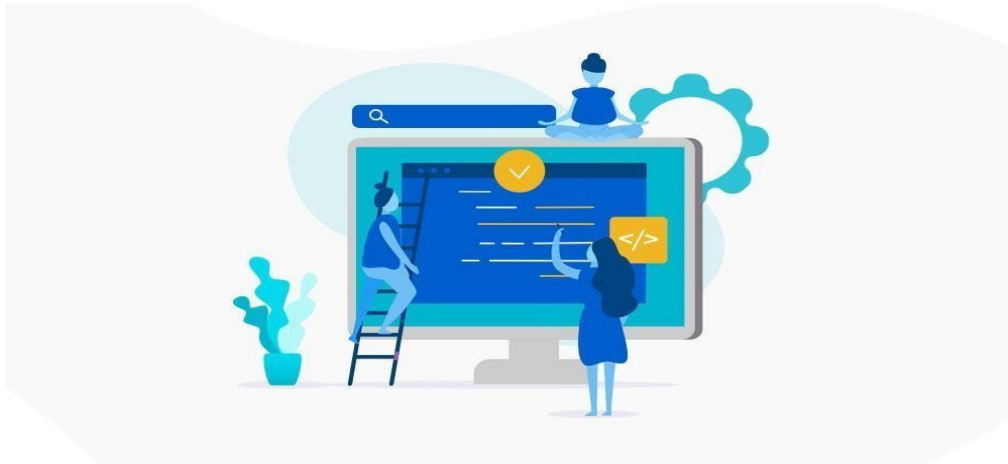


Test Plan helps us determine the effort needed to validate the quality of the application under test. The test plan serves as a blueprint to conduct software testing activities as a defined process which is minutely monitored and controlled by the test manager.

Testing Strategies

A software or QA strategy is an outline describing the software development cycle testing

MASTER BUDGET



approach. The strategies describe ways of mitigating product risks of stakeholders in the test level, the kind of testing to be performed and which entry and exit criteria would apply.

Software testing or Quality Assurance strategies describe how to mitigate product risks of stakeholders at the test level, which kinds of testing are to be done and which entry and exit criteria will apply. They're made based on development design documents.

Types of testing strategies

Some of the testing methodologies that may be part of an organization's testing strategy are:

- † Analytical strategy
- † Model based strategy
- † Methodical strategy
- † Standards compliant or Process compliant strategy
- † Reactive strategy
- † Consultative strategy

† Analytical strategy:

MASTER BUDGET

For example, risk-based testing or requirements-based testing. Here the testing team defines the testing conditions to be covered after analysing the test basis, be it risks or requirements, etc,

So, in case of testing based on requirements, requirements are analysed to derive the test conditions. Then tests are designed, implemented and executed to meet those requirements.

Even the results are recorded with respect to requirements, like requirement tested And passed, those that were tested but failed and those requirements which are not fully tested.

† **Model based strategy:**

In this technique, testing team chooses an existing or expected situation and creates a model for it, taking into account inputs, outputs, processes and possible behaviour. The models are also developed according to existing software, hardware, data speeds, infrastructure, etc.

Let us consider the scenario of mobile application testing. To carry out its performance testing, models may be developed to emulate outgoing and incoming traffic on mobile network, number of active/inactive users, projected growth, etc

† **Methodical strategy:**

Here test teams follow a predefined quality standard (like ISO25000), checklists or simply a set of test conditions. Standard checklists can exist for specific types of testing (like security), application domains.

For instance, in case of maintenance testing, a checklist describing important functions, their attributes, etc. is sufficient to perform the tests

MASTER BUDGET

† Standards or process compliant strategy:

Medical systems following US Food and Drugs Administration (FDA) standards are good examples of this technique.

Here the testers follow the processes or guidelines established by committee for standards or panel of industry experts to identify test conditions, define test cases and put testing team in place.

In the case of a project following Scrum Agile technique, testers will create its complete test strategy, starting from identifying test criteria, defining test cases, executing tests, report status etc. around each user stories.

† Reactive strategy:

Here tests are designed and implemented only after the real software is delivered. So testing is based on defects found in actual system.

Consider a scenario where exploratory testing is being used. Test charters are developed based on the existing features and functionalities. These test charters are updated based on the results of the testing by testers. Exploratory testing can be applied to Agile development projects as well.

† Consultative strategy:

As the name suggests, this testing technique uses consultations from key stakeholders as input to decide the scope of test conditions as in the case of user directed testing. Let us consider a situation where the compatibility of any webbased application with possible browsers is to be tested. Here the application owner would provide a list of browsers and their versions in order of priority.

MASTER BUDGET

They may also provide a list of connection types, operating systems, anti malware software, etc. against which they want the application to be tested.

The testers may then use different techniques like testing pair wise or equivalence partitioning techniques depending upon priority of the items in the provided lists.

† Regression averse strategy:

Here testing strategies focus on reducing regression risks for functional or nonfunctional product parts.

Continuing our previous example of web application, if the application needs to be tested for regression issues, testing team can create test automation for both typical and exceptional use cases.

They can even use GUI based automation tools so that the tests can be run whenever the application is changed.

It is not necessary to use any one of the techniques listed above for any testing project. Depending on product and organization's requirements, two or more techniques may be combined.

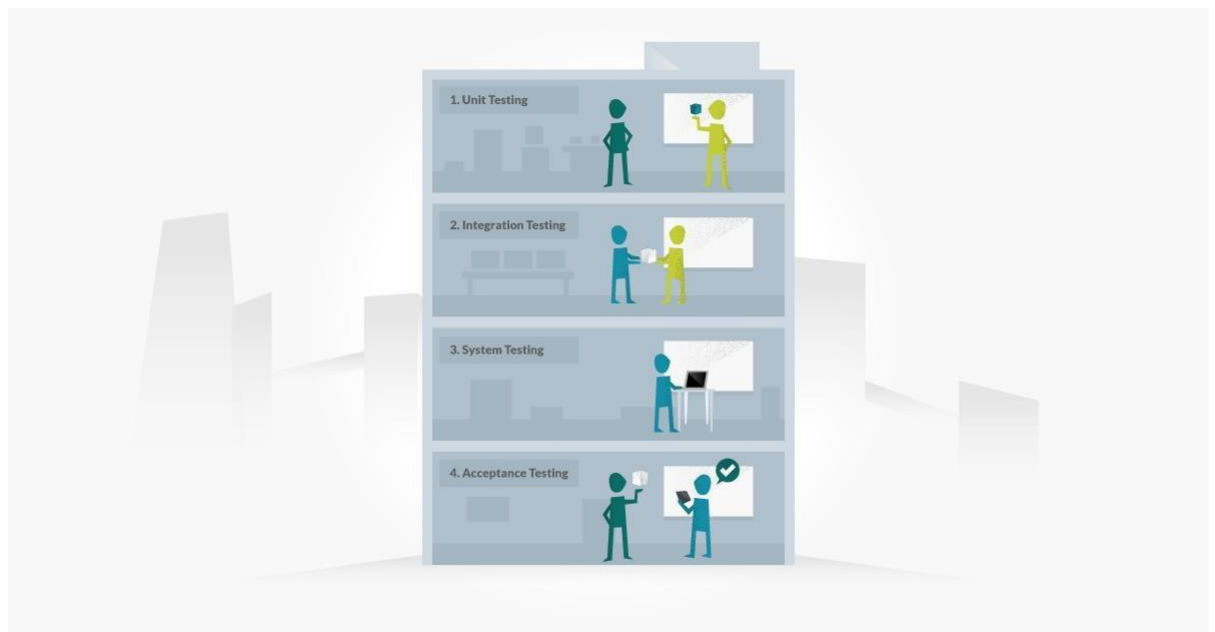
MASTER BUDGET

Level of Testing

The purpose of Levels of testing is to make software testing structured/efficient and easily find all possible test cases/test scenarios at a particular level. In the SDLC model, there are personalized phases such as requirement meeting, analysis, coding, design, execution, testing, and deployment. These all phases go through the process of levels of testing in software testing.

There are many different testing levels that are used or help to check actions and performance for software testing. These testing levels are designed for missing areas and re-coded and re-linked between the development lifecycle states.

In general, mainly four levels of testing in software testing: Unit Testing, System Testing, Integration Testing, and Acceptance Testing.



- ✚ Unit Testing
- ✚ Integration Testing
- ✚ System testing
- ✚ Acceptance testing

MASTER BUDGET

Every testing level is very important testing for software testing but these four levels of testing are very important testing for software engineering.

† Unit Testing:

This type of testing used tests for a single component or a single unit in software testing and this kind of testing is performed by the developer.

Unit testing is also the first level of functional testing. The primary goal of unit testing is to validate the performance of unit components.

Unit is the smallest testable portion of the system or application and the main aim is to test that each individual component or unit is correct in terms of fulfilling requirements and desired functionality.

The main advantage of this testing is that by detecting any errors in the software early in the day is that by doing so the team reduced software development risks, as well as time and money wasted in having to go back and take back fundamental defects in the program once it is nearly completed.

† Integration Testing:

Integration testing means combining different software modules and phases and testing as a group to make sure that the integrated system is ready for system testing or not, and there are many ways to test how different components of the system function at their interface. This type of testing is performed by testers and integration testing finds the data flow from one module to other modules.

MASTER BUDGET

† System Testing:

System testing is most probably the final test to identify that the system meets the specification and criteria and it evaluates both function and non-functional needs for the testing.

System testing is allowing to check the system's compliance as per the requirements and all the components of the software are tested as a whole in order to ensure that the overall product meets the requirements specified. It involves load, reliability, performance, and security testing.

System testing is a very important step as the software is almost ready for production in the market and once it is deployed it can be tested in an environment that is very close to the market/user-friendly environment which the user will experience.

† Acceptance Testing:

The aim of acceptance testing is to evaluate whether the system complies with the enduser requirements and if it is ready for deployment.

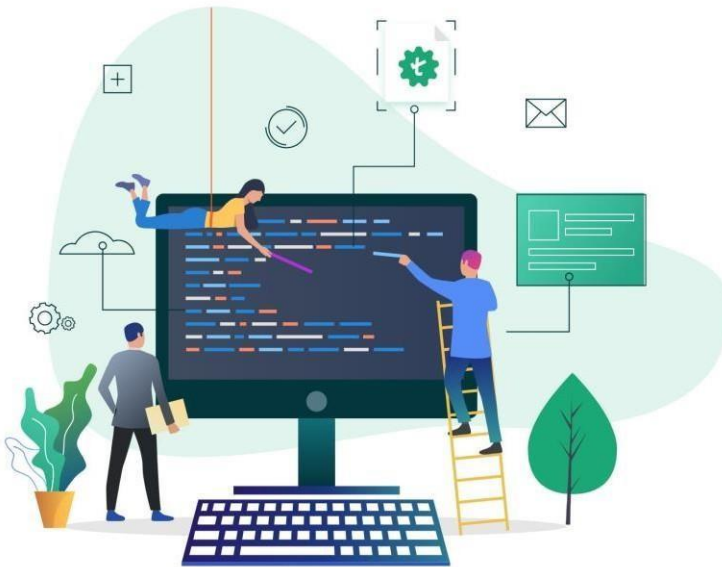
The tester will utilize a different method such as pre-written scenarios and test cases to test the software and use the results obtained from these tools to find ways in which the system can be improved also QA team or testing team can find out how the product will perform when it is installed on the user's system.

Acceptance testing ranges from easily finding spelling mistakes and cosmetic errors to relatable bugs that could cause a major error in the application.

MASTER BUDGET

Types of Testing

Software testing is generally classified into two main broad categories: functional testing and non-functional testing. There is also another general type of testing called maintenance testing.



Functional Testing :

Functional testing involves the testing of the functional aspects of a software application.

When you're performing functional tests, you have to test each and every functionality.

You need to see whether you're getting the desired results or not.

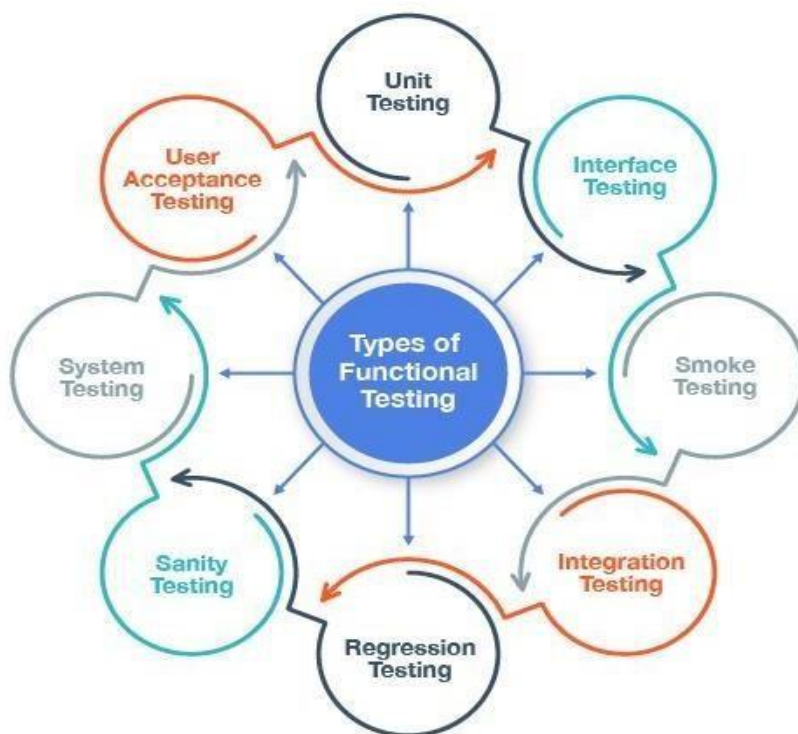
Functional tests are performed both manually and using automation tools.

For this kind of testing, manual testing is easy, but you should use tools when necessary

There are several types of functional testing, such as:

MASTER BUDGET

- ✚ Unit Testing
- ✚ Interface Testing
- ✚ Smoke Testing
- ✚ Integration Testing
- ✚ Regression Testing
- ✚ Sanity Testing
- ✚ System Testing
- ✚ User Acceptance Testing



MASTER BUDGET

Unit Testing- The most basic unit or component of an application is tested usually by the developer to ensure the smallest testable code is working fine. It is a White Box testing technique and appears at the bottom of the V-Model, i.e. it is the first software testing type to be carried out in SDLC.

Integration Testing- When two or more components or units of the software are integrated together, they need to interact with each other in the form of commands, data exchange or DB calls. Integration testing is performed on them as a single cluster to check that the interaction between them is happening, as expected.

Interface Testing- It comes under Integration testing, the correctness of data exchange or transfer between two components is tested in Interface testing. For e.g. One component creates a .csv file as output, the other connected component processes the .csv file into XML and sends to the third component. During this data transfer, the data should remain intact and all the components should be able to process the file and send the file to the next component successfully.

System Testing- All the modules of the application are combined and the whole system is tested as a single unit for correctness against the requirement specification.

Regression Testing- Whenever there are some code fixes, or any functionality enhancement, the code is modified. Regression testing makes sure that these code changes have not injected any new defects in the code and that the previously working functionality is also intact and working.

Smoke Testing- Performed on initial unstable builds, whenever a new build is released by developers, testing team performs Smoke testing to be sure that

MASTER BUDGET

all end-to-end functionalities are working. If any of the major functionality is broken due to a new build, the build is rejected and sent again to developers. This is performed to ensure that new code changes have not broken any major functionality and the build can be taken forward for the next level of testing.

Sanity Testing- It is normally a subset of Regression tests; it is performed whenever there is a new build released for a stable application. Only after running the Sanity test suite, the build is taken forward for the next level of testing. Difference between Smoke and Sanity is-Smoke testing is performed on an initial unstable application, whereas Sanity is performed on a stable application.

User Acceptance Testing- It is the last level of testing in a V-Model, parallel to the requirement analysis phase. It defines how well the application is accepted by the real end- users. The fulfilment of business requirements, by actual users, is checked, in a real environment.

MASTER BUDGET

Testing Methods



† Black Box

† White Box

❖ Gray Box

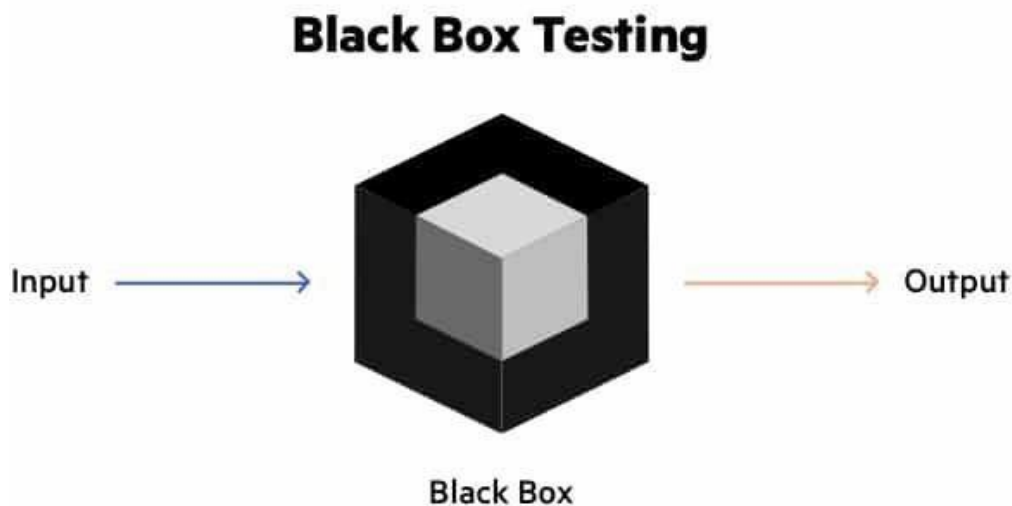
What is Black Box Testing

Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. This makes it possible to identify how the system responds to expected and unexpected user actions, its response time, usability issues and reliability issues.

Black box testing is a powerful testing technique because it exercises a system end-to-end. Just like end-users “don’t care” how a system is coded or architected, and expect to receive an appropriate response to their requests, a tester can simulate user activity and see if the system delivers on its promises. Along the way, a black box test evaluates all relevant subsystems,

MASTER BUDGET

including UI/UX, web server or application server, database, dependencies, and integrated systems. An example of a security technology that performs black box testing is Dynamic Application Security Testing (DAST), which tests products in staging or production and provides feedback on compliance and security issues.



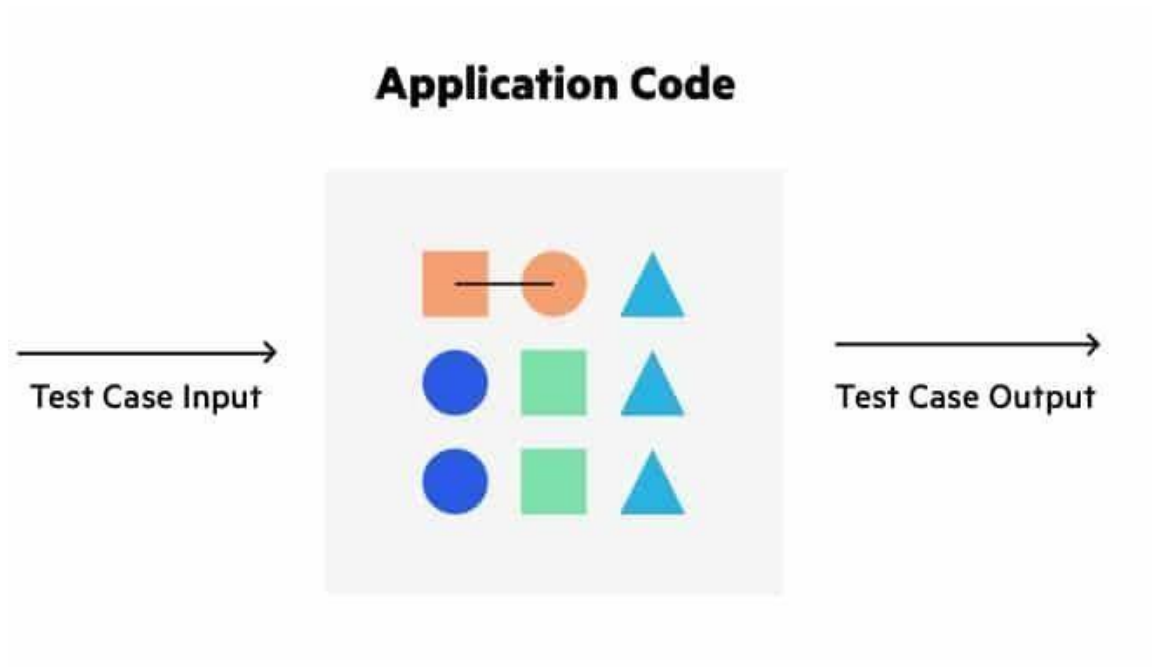
What is White Box Testing:

White box testing is an approach that allows testers to inspect and verify the inner workings of a software system—its code, infrastructure, and integrations with external systems. White box testing is an essential part of automated build processes in a modern Continuous Integration/Continuous Delivery (CI/CD) development pipeline.

White box testing is often referenced in the context of Static Application Security Testing (SAST), an approach that checks source code or binaries automatically and provides feedback on bugs and possible vulnerabilities.

White box testing provides inputs and examines outputs, considering the inner workings of the code

MASTER BUDGET



What is Grey Box Testing:

Grey box testing (a. k. a grey box testing) is a method you can use to debug software and evaluate vulnerabilities. In this method, the tester has limited knowledge of the workings of the component being tested. This is in contrast to black box testing, where the tester has no internal knowledge, and white box testing, where the tester has full internal knowledge.

You can implement grey box testing as a form of penetration testing that is unbiased and non- obtrusive. In these tests, the tester typically knows what the internal components of an application are but not how those components interact. This ensures that testing reflects the experiences of potential attackers and users.

Grey box testing is most effective for evaluating web applications, integration testing, distributed environments, business domain testing,

MASTER BUDGET

and performing security assessments. When performing this testing, you should create clear distinctions between testers and developers to ensure test results aren't biased by internal knowledge.



TEST CASE

In the simplest form, a test case is a set of conditions or variables under which a tester determines whether the software satisfies requirements and functions properly.

A test case is a single executable test which a tester carries out. It guides them through the steps of the test.

You can think of a test case as a set of step-by-step instructions to verify something behaves as it is required to behave.

MASTER BUDGET



A test case usually contains:

- † Title
- † Description
- † Input Data
- † Expected Result
- † Actual Result (once tested)
- † Status

Shree Swaminarayan College of Computer Science

MASTER BUDGET

1. Registration

Test Case No:	1
Title	Registration
Description	This test case allows to register user.
Input Data	Enter user registration detail.
Expected Result	<p>Case 1- In Registration User Can Enter Details (Name, Email-ID, Select Country, Password).</p> <p>Case 2 – If password less then 8 Char/Digits then it will show error message” Please, Enter Strong Password”.</p> <p>Case 3 - If User has entered Email which already exist in database, then it will show error message” Email id already exists”.</p> <p>Case 4 – If Password and Re-password both are different then it will show message “Please, Enter Valid Password”.</p> <p>Case 5 – If Email ID and Password are as per Requirements then it will show message “Account Created ”.</p>
Actual Result	System allowed user registration
Status	Pass.

MASTER BUDGET

2. Login

Test Case No:	2
Title	Login Verification
Description	This test case allows the USER/ADMIN to access the system.
Input Data	Enter valid Username and Password
Expected Result	<p>This test case will check the Username and Password from the Account table in MySQL database</p> <p>Case 1 - If entered Username and Password are match as Admin Details, then it will open admin side page.</p> <p>Case 2 - If entered Username and Password are match from Account Table, then it will open User side page.</p> <p>Case 3 - If Username and Password is entered and doesn't match, then it will show error message "Please Enter Correct Login Details".</p>
Actual Output	System works properly
Status	Pass.

MASTER BUDGET

3. Forget Password

Test Case No:	3
Title	Forgot Password
Description	This test case allows us to change password
Input Data	Enter Email id.
Expected Result	<p>This test case will allow us to change a password.</p> <p>Case 1- In forgot Password if Email id not correct, then it will show error message “Email-ID does not exist in Database”.</p>
Actual Result	System allowed to change password.
Status	Pass.

MASTER BUDGET

4. New Transaction

Test Case No:	4
Title	Add New Transactions
Description	This test case allows to Add New Transactions detail in Transactions
Input Data	Enter transaction details
Expected Result	<p>This test case will allow to check enter all fields to insert into transactions table</p> <p>Case 1- In new transactions add/select the category then it will allow us to submit it.</p> <p>Case 2 – In new transactions all fields are not entered, then it will show error message “Please fill out this field”.</p>
Actual Result	System allows to New Transaction detail.
Status	Pass.

MASTER BUDGET

5. Add Category

Test Case No:	5
Title	Categories
Description	This test case allows to add categories name in add categories
Expected Result	<p>This test case will allow to check whether the user entered categories name or not in Categories Master.</p> <p>Case 1- In Categories user entered the categories name, then it will allow us to submit it.</p> <p>Case 2 - In Categories user does not enter the categories name, then it will show error message “Please fill out all field”.</p>
Actual Result	System allowed to insert categories name.
Status	Pass.

MASTER BUDGET

6. Report Generate

Test Case No:	6
Title	Generate Report
Description	This test case allows to apply filter and generate Report
Input Data	Enter Search value and Date between transactions
Expected Result	<p>This case allows to enter Search value and Date between Transactions for generate report</p> <p>Case 1- In Report enter Search value and Date between transactions,</p> <p>If search value and date find from database then it will generate Report for download.</p>
Actual Result	System generate report
Status	Pass.

MASTER BUDGET

7. Add Investment

Test Case No:	7
Title	Add Investment
Description	This test case allows to add Investment details in Investments by Admin
Expected Result	<p>This test case will allow to check whether the admin enter Investment Details or not in Investment.</p> <p>Case 1- In Investment Admin entered the Investment details, then it will allow us to submit it.</p> <p>Case 2 - In Investment Admin does not enter the Investment details, then it will show error message "Please fill out all field".</p>
Actual Result	System allowed to insert categories name.
Status	Pass.

MASTER BUDGET

Chapter- 5 Conclusion

LIMITATION OF SYSTEM

- **Download Report:** A master budget is support only download Report. That cannot allow to Import/Export any other file. Download only PDF format.
- **Bank Account:** User can add only one bank account.
- **Limited Customization:** Predefined templates may not fit all business needs. Custom reports and formats may not be flexible.
- **Transactions:** User can enter transactions manually.

FUTURE SCOPE OF APPLICATION

- **Mobile App Development:** Expand to Android and iOS platforms for enhanced accessibility.
- **Multiple Currency Support:** Enable transactions in different currencies (**Note:** only static currency).
- **Cloud Integration:** Sync data with cloud services like Google Drive and Dropbox for backup.

MASTER BUDGET

BIBLIOGRAPHY

Learning Resources for Web Development:

- [W3Schools](#) – PHP, HTML, JavaScript, CSS
- [Control Finance](#) – Finance Website for Reference
- [Google](#) – MySQL

This system sets the foundation for a user-friendly and efficient MASTER BUSGET platform, with scope for future improvements to enhance the user experience.