MASTER OF INFORMATION SYSTEMS

Capstone Project

UNIVERSITY OF THE PHILIPPINES
OPEN UNIVERSITY

MASTER OF INFORMATION SYSTEMS

DINIA GRACE D. GEPTE

OPENBANKPH

Thesis Adviser:

RIA MAE BORROMEO
Faculty of Information and Communication Studies

19 December 2023

Permission of the classification of this academic work access is subject to the provisions of applicable laws, the provisions of the UP IPR policy and any contractual obligations:

Invention (I)	□ Yes or ☑ No
Publication (P)	☑ Yes or □ No
Confidential (C)	□ Yes or ☑ No
Free (F)	☑ Yes or □ No

Student's signature:

Thesis adviser signature:

University Permission Page

TITLE

"I hereby grant the University of the Philippines a non-exclusive, worldwide, royalty-free license to reproduce, publish and publicly distribute copies of this Academic Work in whatever form subject to the provisions of applicable laws, the provisions of the UP IPR policy and any contractual obligations, as well as more specific permission marking on the Title Page."

"I specifically allow the University to:

Specifically, I grant the following rights to the University:

a. Upload a copy of the work in the theses database of the college/school/institute/department and in any other databases available on the public internet

b. Publish the work in the college/school/institute/department journal, both in print and electronic or digital format and online; and

c. Give open access to the work, thus allowing "fair use" of the work in accordance with the provision of the Intellectual Property Code of the Philippines (Republic Act No. 8293), especially for teaching, scholarly and research purposes.

Dinia Grace D. Gepte / 19 December 2023

Signature over Student Name and Date

Acceptance Page:

This paper prepared by **DINIA GRACE D. GEPTE** with the title: **"OPENBANKPH: Equipping Philippine Banks in a Modern Digital World"** is hereby accepted by the Faculty of Information and Communication Studies, U.P. Open University, in partial fulfillment of the requirements for the degree Course.

RIA MAE H. BORROMEO	06 February 2024
Adviser	(Date)
BLANCAFLOR P. ARADA	22 February 2024
Program Chair	(Date)

DIEGO SILANG S. MARANAN
Dean
Faculty of Information and Communication Studies

(Date)

Biographical Sketch

Dinia Grace D. Gepte was born in Manila, Philippines. She is a passionate IT professional and global citizen who values excellence. After completing her high school at Evangel Christian School in New York with a Regents diploma, she entered Stony Brook University. She developed a strong foundation in programming and earned a double degree of Bachelor of Science in Computer Science and Applied Mathematics and Statistics, graduating with honors.

To expand her knowledge outside her field, she attended a graduate program at Centennial College in Ontario, Canada, and pursued Global Business Management. During this time, she became a web developer for the Enactus: Pure Success chapter in the institution. In 2015, she became one of the few selected students who represented her school in an exchange program with the Università degli Studi di Milano-Bicocca in Milan, Italy, to apply her expertise in using ICTs to promote sustainable tourism.

Dinia's professional journey includes being a data science engineer, where she honed her skills in developing API-ready back-end systems and data analysis for a financial system. Meanwhile, as an advocate for continuous learning, she entered the Master of Information Systems program at the University of the Philippines – Open University. Her experience with financial technology gave her the inspiration for her capstone project.

Currently holding the position of Senior Specialist – Information Systems (Software Project Manager) at the International Rice Research Institute, Dinia Grace D. Gepte continues to merge academia with practical applications in the IT industry. She also actively participates in professional development opportunities and conferences to stay abreast of emerging trends in the rapidly evolving IT landscape.

Acknowledgement

I extend my heartfelt appreciation to my parents, sister, and relatives for their support, patience, and encouragement throughout this academic pursuit. I would like to thank my professional colleagues for their understanding during the project's implementation. I would also like to express my gratitude to all individuals, particularly Mon, Vanessa, and Joshua, who participated in this study, dedicating their time and providing valuable feedback.

Dedicated to:
My family whose unwavering support
and encouragement have been my pillars,
and to the grace of God, the source of my strength,
guiding me through the completion of this capstone project.

TABLE OF CONTENTS

List of Figures

List of Appendices

ABSTRACT

Despite the growing number of mobile users in the Philippines and the country consistently leading in “time spent on social media” charts, the same cannot be said for the country’s digital banking ecosystem. The lack of mobile financial apps and online banking platforms for the thousands of financial institutions recognized by the Bangko Sentral ng Pilipinas leaves many Filipinos unbanked and prevents many from shifting from face-to-face transactions to digital banking. To help promote financial inclusion through modern technology, a no-cost platform was selected that enabled open banking to simulate a bank with Open API. A mobile-friendly web application was then built using an emerging technology. The result was a usable, functional, and well-performing online app that serves basic banking needs and follows government guidelines. As the system is open-source, it allows hundreds of financial institutions to freely adopt this system to make fast, progressive changes towards digital finance in the Philippines.

Chapter I

THE PROBLEM DOMAIN

Statement of the Problem

It is projected that between 2024 and 2028, mobile phone users in the Philippines will increase by 22.69%, reaching an estimated 59.43 million users [4]. With the projected population of the Philippines in 2028 at 119.1 million [5], nearly 50% of the population is expected to have access to a mobile phone. In contrast to this growth, the BSP has reported that only 8% of adults use their mobile phones for financial transactions, and a mere 5% utilize the internet for the same purpose [6]. Surprisingly, despite the high mobile phone ownership, as of 2021, 41 million Filipinos remain unbanked, representing 53% of the adult population. Astonishingly, 7 out of 10 unbanked adults own mobile phones [7].

An *unbanked* individual is without a bank account or access to traditional banking services. These statistics underscore the prevalent financial exclusion and underservice among most Filipinos. Utilizing technology, such as mobile banking, digital payments, and fintech solutions, presents an opportunity to reduce the number of unbanked individuals.

The COVID-19 pandemic highlighted this potential when many Filipinos turned to e-wallets as their primary mode of payment. However, must we wait for another extreme condition before rapidly changing our financial ecosystem?

Background and Objectives of the Project

In September 2019, the European Union (EU) implemented the Payment Service Directive 2 (PSD2), mandating European banks to provide Open Application

Programming Interfaces (APIs) alongside their IT systems to facilitate online banking services, particularly electronic payment transactions. It marked the second phase of EU legislation designed to break the monopoly of consumer data held by banks, promote transparency, encourage competition, and foster innovation in the financial technology industry [1]. This technology is also known as *open banking*. Investopedia defines it as "a banking practice that provides third-party financial service providers open access to consumer banking, transaction, and other financial data from banks and non-bank financial institutions through APIs. Open banking will allow the networking of accounts and data across institutions for use by consumers, financial institutions, and third-party service providers."

More financial institutions worldwide are embracing this practice, and the Philippines has begun working towards enabling an open banking ecosystem. Following the same pattern as the EU, the Bangko Sentral ng Pilipinas (BSP) published its *Digital Payments Transformation Roadmap for 2020-2023* whose primary goal is to boost payment digitalization in the country. It recognized the benefits of APIs as a standard form of communication between financial institutions, using consent-driven data to enable data portability, interoperability, and collaborative partnerships among incumbent financial institutions and new third-party players. Understanding the associated risks related to data security and privacy, BSP also established the *Open Finance Network of 2021* [3] to lay down the technical, security, and governance standards to be observed by institutions intending to publish open data APIs [2].

By leveraging the data exchanged through open banking, we can anticipate a proliferation of financial institutions and third-party developers publishing diverse software and applications, offering various financial services, particularly within the mobile market–the ideal platform for such advancements.

Significance and Scope of the Project

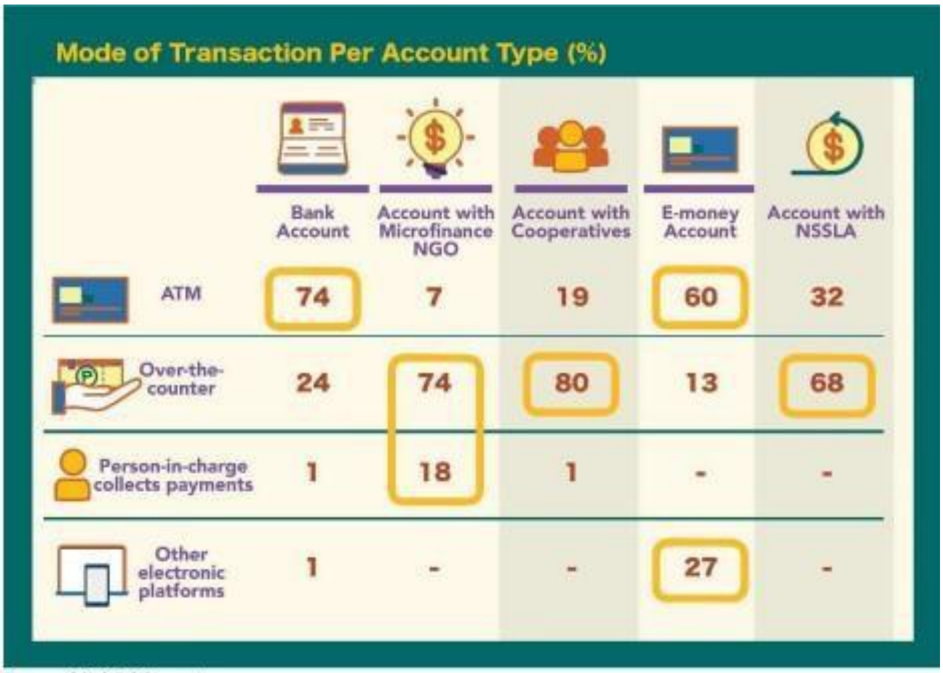
This project was initiated to expedite the readiness of Philippine banks for open banking, leveraging an open-source platform designed to facilitate the opening of bank data. This initiative aimed to offer a rapid and cost-effective solution for banks to prepare their systems for open banking. As a demonstration of Open API capabilities, a basic mobile-friendly online banking application was developed, integrating seamlessly with this API. Furthermore, this solution was designed to be adaptable and customizable, particularly beneficial for smaller institutions seeking to provide their customers with a digital banking experience.

Enhancing accessibility to an Open API-ready online banking system holds the potential to elevate the digital banking landscape for Filipinos. This improvement could attract the unbanked population by reducing the reliance on physical branches for basic transactions, ultimately saving time and costs.

Documentation of Existence and Seriousness of the Problem

The BSP supervises the Philippine banking ecosystem and encompasses 45 universal and commercial banks, 42 thrift banks, 389 rural and cooperative banks, six digital banks, and five non-banks with quasi-banking functions [8].

For banked Filipinos, Figure 1 illustrates that most bank and e-money accounts prefer digital means, primarily through ATMs and other electronic platforms [2]. Conversely, individuals holding accounts with microfinance NGOs, cooperatives, or non-stock savings and loan associations (NSSLA) tend to favor over-the-counter transactions.



Mode of Transaction Per Account Type (%)

	Bank Account	Account with Microfinance NGO	Account with Cooperatives	E-money Account	Account with NSSLA
ATM	74	7	19	60	32
Over-the-counter	24	74	80	13	68
Person-in-charge collects payments	1	18	1	-	-
Other electronic platforms	1	-	-	27	-

Source: 2019 FIS Report

Figure 1 [2]. Mode of Transaction Per Account Type.

Leading Philippine banks offer comprehensive online banking systems via web and mobile applications, enabling users to conduct diverse transactions such as account inquiries, fund transfers, and bill payments. However, rural banks, cooperatives, and smaller financial institutions often lack digital options for their clientele. Many of these entities rely on static websites or Facebook pages, necessitating physical visits for banking services.

Despite the BSP roadmap [2] and framework [3] being published three years ago, none of the systems across the over 400 recognized banks in the Philippines have implemented an Open API, except for Union Bank, Citi Philippines, PayMaya, and Standard Chartered [9].

Significance of the Study

The study aimed to demonstrate how an Open API-ready online banking system could reduce the dependence on physical branches for basic transactions.

This shift could save time and costs for both financial institutions and customers, making banking services more accessible. By leveraging technology, particularly through mobile banking, the study sought to attract the unbanked population in the Philippines. This is crucial in the country where a significant portion of the population remains excluded from modern banking services.

The study contributed to technological advancements in the Philippine banking sector by promoting the adoption of Open APIs. This would encourage innovation and competition among financial institutions, fostering a more dynamic and responsive financial technology industry. The development of a basic mobile-friendly online banking application showcased the possibilities of leveraging open banking for creating user-friendly and innovative financial services.

The study emphasized the development of a solution that is not only rapid but also cost-effective. This is particularly significant for smaller financial institutions that may have limited resources. The study aimed to showcase the adaptability and customizability of the solution for these institutions. Through the adoption of open banking practices, even smaller institutions could offer digital banking experiences to their customers, enhancing their competitiveness in the market.

The study aligned with the BSP’s Digital Payments Transformation Roadmap. It served as a practical initiative to implement the objectives set forth by BSP in boosting payment digitalization and establishing technical, security, and governance standards through the Open Finance Network. The project demonstrated the feasibility of adopting open banking practices, providing a tangible example for other financial institutions to follow. This aligned with the broader goals of the BSP in transforming the digital payments landscape in the country.

Scope and Limitation of the Study

The study operated within the existing regulatory framework established by the BSP. Adherence to these standards could restrict the extent to which the open-source platform and the associated online banking application can deviate from established guidelines. Changes in regulatory requirements or unforeseen challenges could impact future work.

The integration of the open-source platform with diverse banking systems could pose technical challenges, and encounter difficulties in ensuring seamless communication and interoperability between different banking IT infrastructures. Technical constraints may arise due to variations in technology stacks, security protocols, and data structures among different banks. These compatibility issues could affect the scalability and applicability of the proposed solution for implementing groups.

The development and implementation of the online banking application was limited to a specific time frame, which impacted the depth and breadth of the study's outcomes.

Operational Definition of Terminologies

A. Open Banking

Open banking refers to the banking practice that mandates financial institutions to provide Open Application Programming Interfaces (APIs) alongside their IT systems. These APIs facilitate third-party access to consumer banking, transaction, and financial data, promoting transparency, competition, and innovation in the financial technology industry.

B. Open Application Programming Interface (API)

An Open API is a set of rules and protocols that allows external developers and third-party service providers to access and interact with the data and functionalities of a financial institution's IT systems. It enables secure and standardized communication between different systems.

C. Financial Inclusion

Financial inclusion refers to the extent to which individuals and businesses, especially those in underserved or unbanked populations, have access to and actively use a range of financial services, including banking, digital payments, and other financial products.

D. Unbanked

An unbanked individual is someone who does not have a traditional bank account or access to mainstream banking services. In the context of this study, it specifically refers to individuals without access to digital banking services.

E. Digital Payments Transformation Roadmap

The Digital Payments Transformation Roadmap, as outlined by the Bangko Sentral ng Pilipinas (BSP), refers to the strategic plan and guidelines designed to boost payment digitalization in the Philippines. It includes initiatives, policies, and standards aimed at advancing the adoption of digital payment methods.

F. Open Finance Network

The Open Finance Network, established by the BSP in 2021, serves as a framework defining technical, security, and governance standards. It provides guidelines for financial institutions intending to publish open data APIs, ensuring a secure and standardized approach to open banking practices.

G. Mobile Banking

Mobile banking involves the use of mobile devices, such as smartphones or tablets, to perform various banking activities, including account inquiries, fund transfers, and bill payments. It allows users to access banking services on-the-go through dedicated applications or mobile-friendly websites.

H. Fintech (Financial Technology)

Fintech refers to the use of technology to deliver financial services and products, often leveraging innovative solutions, software, and applications.

I. Online Banking Application

An online banking application, as demonstrated in this study, is a software solution accessible through the internet or mobile devices. It enables users to conduct various banking transactions, such as account inquiries and fund transfers, in a digital and user-friendly manner.

Chapter II

REVIEW OF EXISTING ALTERNATIVES

When considering the addition of an Open API layer to an existing banking system, two prominent options are available: Open Bank Project (OBP) and Apigee Open Banking APIx by Google.

OBP is a semantic API solution tailored for banks and the financial services industry, providing customers access to a third-party app and service ecosystem. It offers 270 ready-to-use APIs, a secure API Gateway, an API portal, a sandbox, core-banking connectors, and robust authentication workflows. In contrast, Apigee is an API management platform that empowers customers to design, secure, analyze, and scale APIs anywhere, specifically offering PSD2-compliant APIs. Its functionalities include API proxies for authenticating and securing financial communications, a developer portal, and an OAuth security framework.

Both solutions possess similar features, yet a significant difference lies in their licensing models. OBP operates on an open-source basis, offering publicly accessible source code. This open-source characteristic renders the product highly extensible. Moreover, the commercial license OBP requires for developing connectors between an existing system and the API framework is likely more cost-effective than a fully commercial product such as Apigee. Additionally, adopting Apigee implies a substantial commitment to Google as the vendor, potentially leading to vendor lock-in for adopters.

Although the BSP [2] recommended the creation of a Central API Sandbox to deploy and test Open APIs within specified parameters and timeframes, no such sandbox environment has been developed to date. Regarding the four banks [4] with

Open APIs, it is highly probable that their APIs were developed in-house from scratch.

In terms of available applications, this project doesn’t intend to replace existing mobile banking apps but seeks to make online banking systems accessible to smaller banks. The major banks in the Philippines have their own apps, such as BDO Online, LANDBANK Mobile Banking, BPI, Metrobank App, etc.

In terms of available third-party apps for consumers to perform banking transactions, the notable options are GCash, Maya, and Spendee.

Maya (previously PayMaya) was among the early apps that offered e-wallets and payment services. Users can top up their wallets with money from other banks through PayMaya’s platform. The wallet funds, like a credit card, can be utilized for online bill payments or purchases. Customers can also use the wallet for payments through point-of-sale systems.

GCash functions similar to Maya, but neither is a comprehensive one-stop-shop solution for banking transactions. Although the apps have evolved to allow users to connect their banks seamlessly, this capability is currently limited to major financial institutions and only facilitates the movement of money between them. Unfortunately, there is no way to view balances for each account within the app without logging into an external application provided by the host bank.

The less popular Spendee is a web and mobile app that connects to multiple personal bank accounts, offering an interface to track finances. Its basic functionalities include shared wallets, a budgeting tool, wallet overview, scheduled transactions, wallet transfers, data import/export, bulk editing of transactions, and notifications. Spendee currently connects with several financial institutions in the Philippines, including BPI, BDO, Citibank, MetroBank, and PayPal.

Chapter III

APPROACH TO BE TAKEN IN THIS PROJECT

Overview

The primary objective of this project was to provide Philippine banks, especially smaller ones, with a low-cost and easily deployable web application linked to an open API-ready banking system. The main deliverables of the project included:

1. A recommendation for an API management tool to expose financial banking data in accordance with open banking standards.
2. Development of a mobile-friendly application designed to replicate an online banking platform for consumers. This application served as an example that any banking institution could adopt to meet basic banking needs.

The system created through this project demonstrated how any app developer, including third parties, could seamlessly interact with a banking system equipped with an Open API.

The web application primarily featured a unified view tailored for bank customers. Its operation involved a combination of client and server-side logic for data processing and database access. The application utilized Open Bank Project's (OBP) APIs to access bank-specific data and transactions, which interfaced with a bank's core IT system. While the project utilized OBP's Sandbox API environment to simulate banking data, it didn't directly handle the setup of OBP's connectors to an actual bank's core system. Banks intending to adopt this solution in the real world would need to acquire a commercial license from OBP or fork out its open-source code to establish the necessary connectors before deploying the web application.

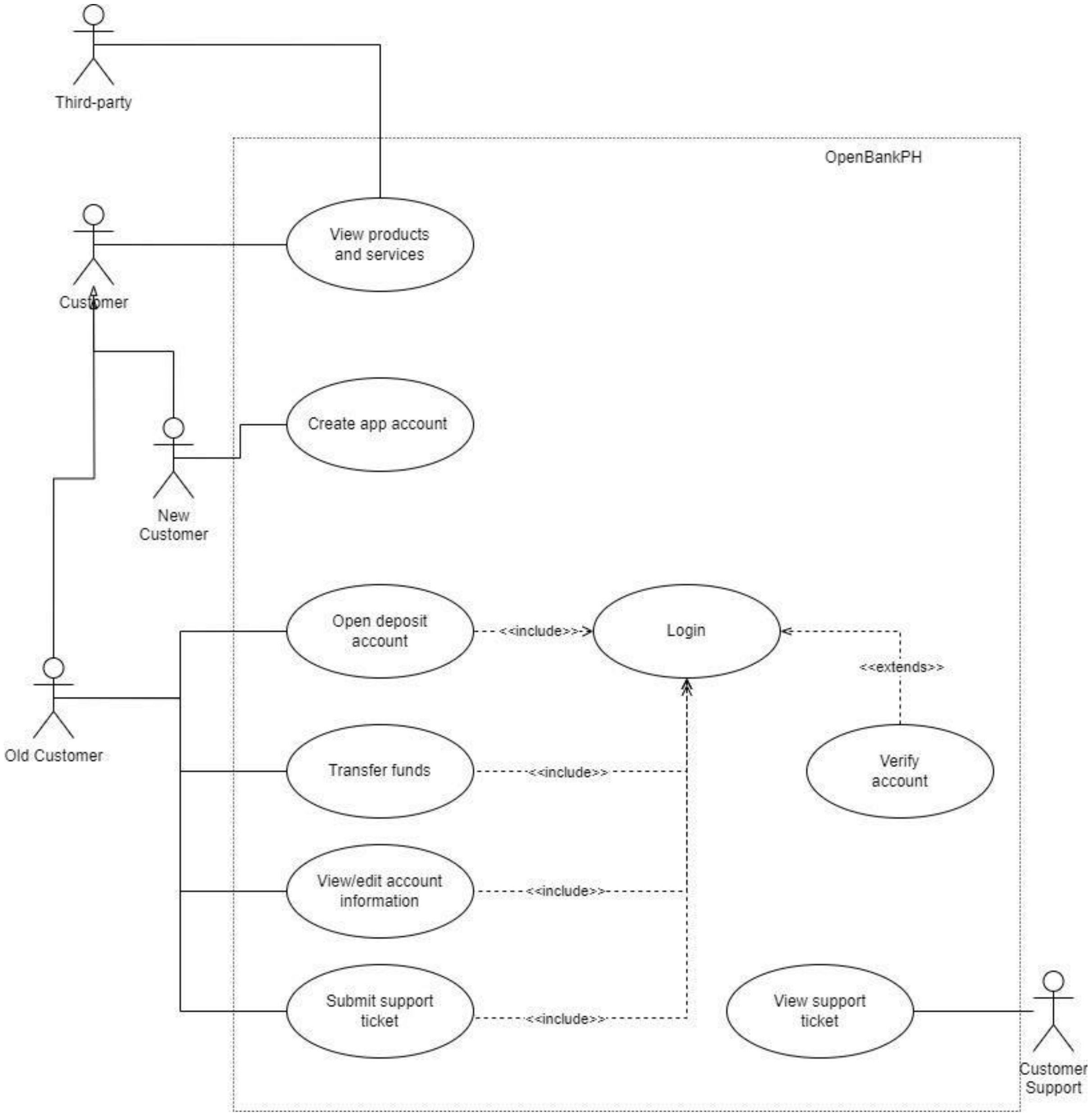


Figure 2. Use-Case Diagram.

Figure 2 illustrates the three distinct user types can interact with the system:

1. **Customers.** These users can be classified as new or old customers. Any new or returning user can view a bank's products and services, as this information is publicly available. New customers can create an account within the web application by providing a valid email address, which requires verification for access to additional features. Once

verified, they become recognized as returning or old customers. Upon login, old customers can perform various actions, including opening new deposit accounts, transferring funds between accounts, managing account information, and submitting support tickets.

2. **Admin.** Customer support staff falls under this category, which the bank typically employs to handle user-submitted inquiries or concerns regarding the bank or its system. This project's support tickets are accessed in their raw format directly through the database using Google Firestore. Other admin operations, such as API and data management, could be accessed through the application's developer dashboard or OBP's API Manager tool.
3. **Third-party.** These entities, such as other bank systems or fintech companies, aim to exchange information with this bank's API. Their interaction occurs solely through the API and not directly with the application's interface.

Theoretical Framework

An application accessible through a web browser or mobile requires a means of communication between clients and servers. This project placed a significant emphasis on Open API-ready systems. Systems utilizing APIs, particularly RESTful ones, rely on a defined set of standard HTTP requests and responses between clients and servers. These requests consist of four main elements: endpoints (URL), methods, headers, and data (or content body).

The *method* of a request identifies the type of operation that must be performed on the server. These operations follow a concept like CRUD (create, read, update, delete) principles used in databases. However, instead of acting on records, the operations are executed on specific resources–pieces of data. These resources align with HTTP POST, GET, PUT, and DELETE methods, respectively, reflecting the universal understanding of these operations in the context of APIs.

Rationale for the Framework

Even after two decades, REST API remains the standard for today's service app development. Alternatives to REST include GraphQL, Falcor, and gRPC, among others. While REST APIs can be tedious and error-prone to implement and use, especially when dealing with one resource at a time, they are widely used, and many developers are familiar with them. One of the strengths of REST lies in its simplicity and ease of use. Accessing someone's REST API doesn't require additional setup from the caller besides acquiring proper authentication; it can be done through web browsers (e.g., Swagger UI), clients (e.g., Postman, Insomnia), or command-line tools (e.g., cURL). This aligns with the goal of making bank information easily accessible and secure for third parties and other companies.

Furthermore, the complexity of bank resources and requests doesn't warrant the need for tree or graph-type models like Facebook's GraphQL. REST continues to be suitable for these scenarios.

In the development of the user-facing application, a low-code approach was employed. According to IBM [11], low-code is “a visual approach to software development that enables faster delivery of applications through minimal hand-coding. [It] enables less technical employees to make a larger business impact in numerous ways, such as relieving IT department backlogs, reducing shadow IT, and taking more ownership over business process management (BPM) workstreams. [It] also aid more seasoned programmers … and shorten the application development lifecycle, enabling them to accomplish more in less time.”

Low-code facilitated rapid prototyping and development of the application. Furthermore, it enables non-programmers to test and utilize the developer dashboard efficiently with minimal coding experience for future iterations.

Technologies Used

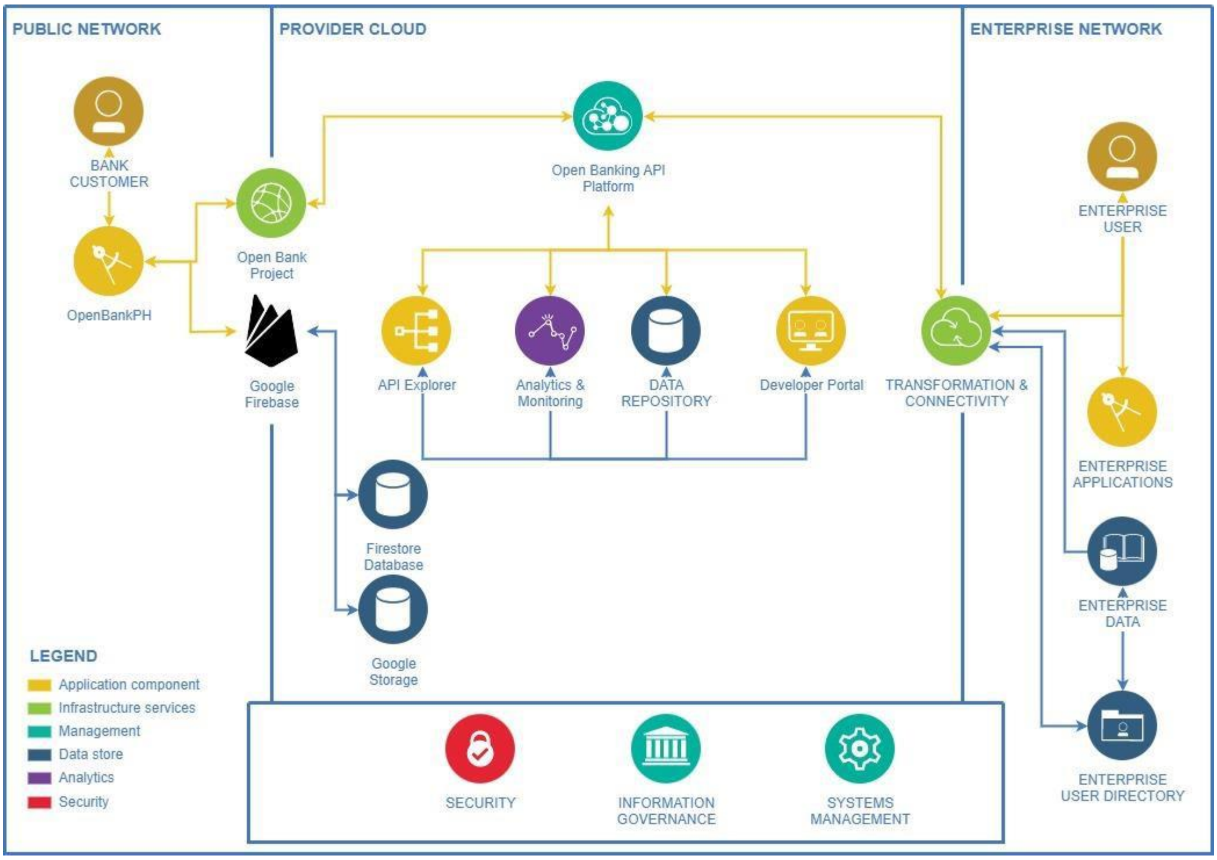


Figure 3. System Architecture.

The client-side code was developed using FlutterFlow, a form of low-code platform. While FlutterFlow offers an array of pre-built functions in the form of drag-and-drop elements, very similar to content management systems, certain parts of the application require custom functions to be written in dart.js. It's also crucial to highlight that any additional package or framework used to develop code functions was open-source, as the project repository is publicly available and extendable for adoption by financial banks without requiring commercial licenses.

Figure 3 depicts the communication line between the web application and a Theoretical bank system's API, using OBP's API. The OBP's Sandbox API environment, operating on version 5.1.0, was injected with dummy data through POST and PUT API requests with JSON payloads. The main resources used in this API were banks, branches, ATMs, customers, users, products, accounts, and transactions. These entities are all stored within the Open Banking API Platform. Furthermore, the OBP platform has connectors to any system to expose data and services, which was not explored during the development of this project.

Google's Firebase and Storage, a NoSQL database and object storage solution, were used for application-side data management specific to the web application. Additionally, Firebase was responsible for handling authentication processes within the application.

For production code deployment, instead of FTP transfers of source code to server hosts or fully managed container systems, FlutterFlow has a fully managed system and acts as the host of the application.

Chapter IV

CHAPTER PLAN

Concept

A. System Design and Features

- **Bank, Branch, and ATM Information.** A public user view of bank information, including website, contact numbers, list of branches and ATMs and their addresses, hours of operation, and other descriptive information.
- **Bank Products.** A public user view of bank products such as deposit accounts and loans, along with their description, product website, maintaining balance, rates, fees, and other charges.
- **Contact.** A bank customer view of the email, contact number, and bank website. It also has a form for submitting support tickets and relevant attachments to help customer support resolve queries.
- **Account Registration.** New customers may create an account within the application using a valid email address. Users need to provide their first name, last name, and at least a 10-character password containing at least one of the following: digit, uppercase letter, lowercase letter, and unique character. By default, new accounts do not contain any product accounts.
- **Secure Access.** Google Firebase secures authentication through the web application, and an authentication token is required to access information in the OBP API. This token is requested upon user login. Bank customers can end their session by clicking on the log-out button of the application. An OBP account with privileges granted through OBP’s Developer Portal was used to develop the application and access API resources.
- **Profile Management.** A bank customer can view and update their personal

details recorded in the system. These are displayed name, age, title, and profile picture.

- **Password Management.** A bank customer can change their login password.
- **Account Summary.** By default, a bank customer is directed to an overview page of their accounts upon login. The overview page displays the number of deposit and loan accounts they own. Each type of product can be further clicked to display account details such as the account name, number, and outstanding balance.
- **Transaction History.** A bank customer can click on their deposit or loan accounts in the Account Summary page, and a list of their account transactions will be displayed. The transaction history contains each transaction's type, description, time, and amount.
- **Bank Account Application.** A bank customer can apply for any type of available deposit account within the application by providing their personal, family, and education & employment information. This feature is only available when the user has verified their email address.
- **Fund Transfer.** A bank customer can transfer funds between their deposit accounts applied through the application. This feature is only available when the user has verified their email address.
- **Bank Product Management.** An admin can create, update, and delete bank products using an OBP account with corresponding credentials through the OBP API Manager. These bank products are visible on the UI.
- **ATM/Branch Info Management.** An admin can create, update, and delete bank ATMs/branches using an OBP account with corresponding credentials through the OBP API Manager. These ATMs/branches are visible on the

UI.

- **Cross-platform and Responsive Design.** The application is accessible via a web browser but was designed with mobile users in mind. It’s deployable through Google Play Store and Apple App Store, provided publishing requirements are met.
- **Light/Dark Mode.** The application can be switched freely between light and dark mode.

B. Database Design

The application-side data was managed using Firestore Database [12], a NoSQL, document-oriented database. It has no tables of rows, and data is stored in *documents* organized into *collections*. Figure 4 describes a *user's* collection within Firestore, with each document referenced by a unique document ID (UID). Additionally, an *accountList* document was created to house a user’s essential banking information required to access data within the API. Another document, *tickets*, was designed to contain information related to user-submitted support tickets. This included a photo URL referencing a Google Object Storage key, where all user uploads are stored.

The OBP database stores data related to bank operations, accounts, and transactions. Developers cannot access the database design; instead, they utilize the API Explorer for CRUD (Create, Read, Update, Delete) operations.

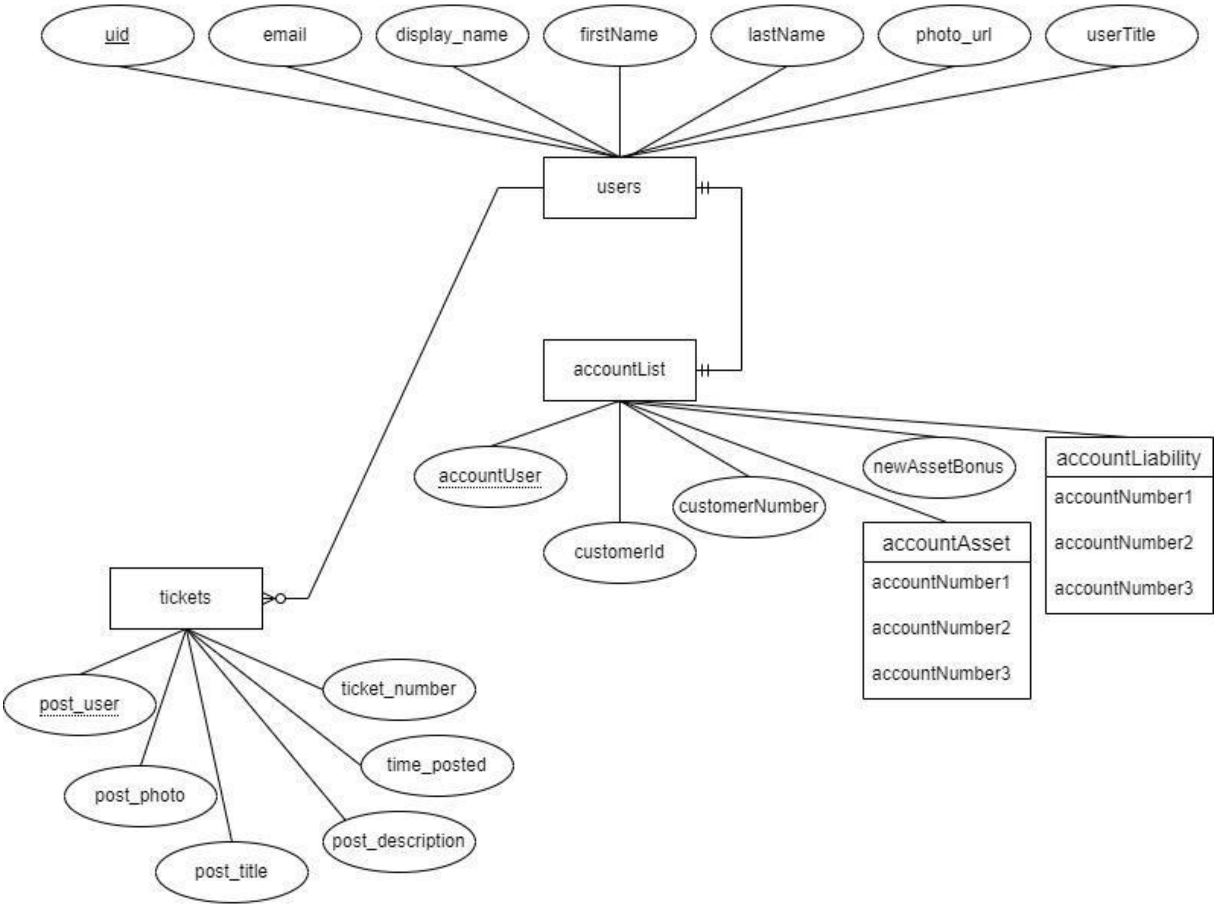


Figure 4. Firestore Database Design.

Methods

In 2020, preliminary research commenced by identifying a gap in existing banking systems: the absence of Open APIs in Philippine banks. This led to the recognition of another issue–limited online banking capabilities among smaller banks and financial institutions. The project's deliverables aimed to tackle both challenges. Development of the application took place during the weekends and some holidays from October to December in 2023.

Initially, the plan was to create an API banking system from scratch. However, upon reviewing existing systems, a widely used and well-maintained open-source API solution, Open Bank Project (OBP), was discovered. The decision was made to incorporate it into the project, opting not to reinvent the wheel. This choice allowed the project to focus on creating an information system that adds more value and has practical applications in actual systems.

OBP offers an added feature–an API sandbox allowing developers to test its integrated APIs. An OBP account was requested via the API Development Portal to access and manage data within the sandbox API. The API Explorer served as a comprehensive tool containing all available API endpoints, featuring example requests and responses. This tool was crucial in creating theoretical entities such as bank branches, ATMs, and products. The Guagua Rural Bank, an official rural bank recognized by the BSP, was utilized to populate the API data to create a more realistic simulation of a banking environment.

While some API resources were accessible using a generic account, the majority required specific permissions granted to the account. The management of these permissions was facilitated through OBP's API Manager tool.

Development using FlutterFlow requires initial setup which is comprehensively documented by the platform. It integrates with Google Firebase to manage user authentication and other application data, so a Google Firebase account was necessary. FlutterFlow provides templates that expedite development; however, becoming acquainted with the platform’s concepts, such as widgets, actions, components, pages, and interactions, necessitated reading their corresponding documentation. Additionally, certain functionalities needed by the application, including data transformations or validations, demanded custom-written code. Code Copilot supplemented this by providing AI assistance in code development.

Enabling the application to interact with OBP endpoints necessitated the configuration of individual HTTP request calls within FlutterFlow. More than 20 endpoints were selected from the OBP API catalog and integrated into the application, most of which required some form of authentication to perform CRUD operations on the sandbox database.

Before deployment onto the live production server via FlutterFlow, the system underwent functional and end-to-end testing. Furthermore, a usability survey was conducted to assess the system's user-friendliness. The survey results would guide future revisions aimed at enhancing the system's usability for adopting institutions.

Plan for User Testing and Project Assessment

A. User Testing

As the project was designed as a template solution for banks lacking an online banking system, the ideal scenario involved partnering with a financial institution to implement the project’s deliverables. Metrics such as customer feedback, website traffic, and system uptime would have been crucial in assessing the project's success. However, due to the absence of a partner bank, a strategy involving adherence to government guidelines, rapid prototyping, manual and automated testing, and a usability survey was employed to assess the system.

The assessment was guided by the BSP guidelines to assess the suitability of the Open Bank Project as an API management and connector tool for exposing bank data. In this evaluation, the Open Finance Framework was referred to, which categorizes the standards for open finance adoption into five tiers [3]:

Tier 1 - Product and Service information. Refers to “read-only” information on financial data and other details of financial products/services that are readily accessible online and can be freely used, reused, and redistributed by any entity, such as deposit/lending rates, credit card offerings, service charges, and other public data.

Tier 2 - Subscription and New Account Applications. Includes customer acquisition and account opening processes, facilitating digital application and submission of supporting documents. Applications for deposits, loans, debit/credit cards, and other financial products are covered under this category.

Tier 3 - Account Information. Refers to personal (e.g., name, registered address, phone numbers, etc.) and financial information provided by a customer at any given time or other details of the account of authenticated customers for stand-alone or aggregated views. This consists of, but is not limited to, data types such as account balance, credit card outstanding balance, transaction records, credit limit change, and credit score.

Tier 4 - Transactions. Covers payments and other financial transactions, such as scheduled payments and transfers initiated by customers.

Tier 5 - Others. Covers other more complex financial products or use cases and those not covered by tiers 1-4.

The application used OBP’s catalog of APIs to perform various functions, including viewing products and services, creating new deposit account applications, managing account information, and transferring funds between accounts. These features indicate that OBP can support Open Finance Standards specified by the government of the Philippines.

A combination of software testing and user feedback was employed to evaluate the web application's usability. Testing within FlutterFlow occurred in a Debug environment where the application was deployed. The software was run in debug mode, where a preview of the application was available, and end-to-end functionality on a per-page basis was conducted. This included testing authentication processes and data modification within their respective databases. FlutterFlow's early-detection system facilitated the easy identification of bugs just before deployment. It notified the developer of warning or error-level bugs. Additionally, custom functions provided an interface for scripting test cases and data input. During manual testing, run-time bugs primarily consisted of UI-related issues, notably elements not fitting the available screen dimensions or data being populated incorrectly.

After the initial deployment to production, the developer carried out a comprehensive end-to-end manual testing to assess all facets of the application. However, an issue with Cross-Origin Resource Sharing (CORS) hindered the functionality of components requiring access to the OBP API. To address this, the solution involved encapsulating the API calls within Firebase Functions instead of directly invoking them within the application. Subsequently, the application underwent a second deployment to production, followed by another round of end-to-end tests to ensure the resolution of the CORS issue and overall functionality.

Between December 2nd and 7th, a feedback form and the application link were distributed to approximately 10-15 respondents of different ages and professional backgrounds. Out of these, six respondents, aged between 24-34, provided feedback. Among them, two-thirds were female, and one-third were male, representing various occupational fields such as Creative, IT/Technology, Healthcare, and Engineering. Notably, the respondents were regular users of mobile apps in their daily routine, and 50% of them held accounts with a Cooperative or Rural Banks. The respondents were users of financial apps such as Maya, GCash, BDO/BPI/AUB/UnionBank, GoTyme, Pag-ibig, CIMB, and Seabank. Among these apps, they highly valued user-friendliness and seamless integration with other platforms. However, some respondents reported challenges with certain apps, citing difficulties in navigation, limited features, slow performance, and frequent crashes.

The project application underwent assessment across usability, functionality, and performance. Respondents generally found the app easy to navigate, encountering only minor difficulties but overall without significant issues. Most agreed that menus and options were clear and largely intuitive, suggesting some areas for improvement. While the majority of features worked well, there were isolated instances of minor glitches or issues with certain functionalities. Although no bugs were encountered, a few users faced technical problems when using the app on an iPhone browser. Despite these minor concerns, the application received an excellent performance rating. Users noted its exceptional speed and instantaneous responsiveness to all interactions. Moreover, the app remained stable throughout usage, demonstrating no instances of crashing or freezing.

B. Security Testing

Running Zed Attack Proxy's (ZAP) automated scan using an Ajax spider with Chrome against the application yielded the following results:

Medium Alerts.

- 10038 - Content Security Policy (CSP) Header Not Set (5)
- 10098 - Cross-Domain Misconfiguration (8)
- 10020 - Missing Anti-clickjacking Header (2)

Low Alerts.

- 10017 - Cross-Domain JavaScript Source File Inclusion (3)
- 10035 - Strict-Transport-Security Header Not Set (11)
- 10096 - Timestamp Disclosure - Unix (26)
- 10021 - X-Content-Type-Options Header Missing (8)

Informational Alerts.

- 10027 - Information Disclosure - Suspicious Comments (21)
- 10109 - Modern Web Application (2)
- 10015 - Re-examine Cache-control Directives (2)

No major risks were found in the application; however, there were a few medium and low alerts. It’s crucial to thoroughly review each vulnerability to identify and understand their impact on the system. Critical vulnerabilities need immediate attention, as they can potentially lead to significant breaches. While the application didn’t have significant issues, it can be used as-is and subjected to further scrutiny by future developers.

Chapter V

DISCUSSIONS

Financial inclusion can be attained through various means, including enhancing accessibility affordability, offering tailored products, promoting financial literacy, driving technological innovation, and refining regulatory frameworks. Among these, this paper primarily emphasized technological innovation and regulatory frameworks. However, despite the government's advancements in technology and regulations, educating Filipinos about the advantages and convenience of online banking in contrast to traditional in-person transactions is vital for driving further growth and adoption of digital solutions. This necessitates full participation and adaptation from financial institutions to align with the evolving financial landscape.

Currently, the majority of Philippine banks do not have publicly available APIs, but PayMongo's CEO sheds light on how [10] third-party integration works presently in the Philippines: "When you want to launch something online for a payment gateway, you have to deal with banks and many different financial institutions. It takes months, we tried it ourselves, from negotiating rates to submitting paperwork. It takes a long time, and then in the end, you are charged high fees". Having an Open API system in banks would simplify the process for third parties, making it faster, easier, and more cost-effective, potentially eliminating the need for exorbitant fees to be passed on to end-users.

As a developer, integrating the application with Open Bank Project presented both benefits and challenges, with the former significantly outweighing the latter. Leveraging OBP’s Sandbox API facilitated rapid prototyping and streamlined the development of an online banking application. While acquiring a developer account was straightforward, initial account permissions posed challenges, particularly in accessing specific API endpoints, notably the creation of a bank entity. This hurdle was resolved through direct communication with API developers based in Europe. OBP's API Explorer lacked comprehensive documentation, requiring some guesswork and prior experience in API development to navigate effectively. For instance, an endpoint designed to retrieve all of a bank's products didn't yield the expected output. It was discovered that the latest version (5.1.0) did not return these values, whereas version 2.0.0 did. Despite encountering challenges, OBP demonstrated exceptional customer support by promptly addressing queries within a 24-hour window, a service extended even to free users. Their swift and knowledgeable responses reflect their expertise within the financial ecosystem and their understanding of their product. This level of support suggests that future adopters can anticipate similar, if not enhanced, customer service when preparing their systems to be Open API-ready.

Utilizing low code for building web applications proved a novel yet challenging experience. The trade-offs between ease of use and control in low-code development became evident. Its user-friendly dashboard allowed for speedy development, enabling anyone with minimal coding experience to create usable applications. Undoubtedly, it could reduce costs by streamlining application development time and minimizing the need for highly skilled developers. However, low-code comes in many platforms. In this project, FlutterFlow with an educational license account was sufficient to build the web application to deployment. Extensive reference to FlutterFlow's documentation and tutorials was crucial throughout development. Towards the end of development, some lags in the developer dashboard were encountered, possibly due to many elements and artifacts. As the application grows, it's conceivable that the developer interface may face performance issues. Another notable concern relates to the control over code. What might be succinctly written in a few lines of code within a traditional coding environment could entail multiple loops and various elements in low-code platforms like FlutterFlow. This intricacy could potentially impact optimization in certain scenarios.

Chapter VI

CONCLUSION

Government guidelines, investment in innovative financial technologies, and prompt action could significantly reduce the number of unbanked individuals, empowering more Filipinos to access and benefit from formal financial services, thereby creating a more robust and inclusive financial ecosystem.

By adopting open banking through Open APIs, financial institutions could contribute to the growth of the digital landscape in the Philippines, allowing for rapid development of fintech applications, notably in the mobile market. To expedite the creation of these APIs, the Open Bank Project stands out as a recommended solution, providing a platform to connect existing bank systems' data with third-party providers seamlessly. It has various tools to aid in the speedy development of such systems, including a Developer Portal, API Explorer, and API Manager. This cost-effective solution holds the promise of granting smaller banks the opportunity to establish their APIs and pave the way for their own online and mobile banking services.

With the advent of open finance in the Philippine banking sector, a surge in diverse applications, including those rapidly developed using low-code and AI, is anticipated to flood the mobile marketplace. This phenomenon mirrors the maturation observed in such systems, notably in Europe, and underscores their beneficial influence on the financial industry. Empowering Philippine banks with the tools necessary to construct their APIs and dynamic web applications marks the initial stride toward embracing this transformative change.

Chapter VII

FUTURE WORK

Financial institutions, regulatory authorities, global development companies, and fintech could use the OBP platform. The government could explore the platform to create the suggested Central API Sandbox in [3]. Third-party developers could create budgeting-like software that allows multiple account management across different banks within a single app.

Furthermore, conducting in-depth research into the capabilities of the Open Bank Project, particularly its connectors to real-world banking systems, is advisable for future endeavors. It is imperative for adopting institutions, however, to ensure that their final API product aligns with the standards set forth in the Open Finance Framework published by the government.

Based on feedback received from user testers and due to constraints on development time, several enhancements could be implemented within the web application. First, ensuring compliance with the Americans with Disabilities Act (ADA) regarding the color palette used in the application is essential. Additionally, integrating a review process into new account applications would provide a safeguard to banks. Incorporating bank-specific products like loans within the application would further enrich its offerings. Creating an interface tailored for non-customers, like customer support functionalities, could enhance usability. Moreover, deploying the application to mobile app stores would expand its reach and accessibility. Lastly, integrating third-party functionalities that enable payment capabilities would augment the application's overall utility and user convenience.

REFERENCES

[1] Group, T. (2020) *PSD2 regulation and compliance - get ready with Thales*, *Thales Group*. Available at: https://www.thalesgroup.com/en/markets/digital-identity-and-security/banking-payment/digital-banking/psd2 (Accessed: May 2020).

[2] *An efficient, inclusive, safe and secure digital payments ecosystem ...* (2020) *Bangko Sentral ng Pilipinas*. Available at: https://www.bsp.gov.ph/Media_And_Research/Primers%20Faqs/Digital%20Payments%20Transformation%20Roadmap%20Report.pdf (Accessed: October 2023).

[3] *Open Finance Framework* (2021) *Circular No. 1122, Series of 2021*. Available at: https://www.bsp.gov.ph/Regulations/Issuances/2021/1122.pdf (Accessed: October 2023).

[4] *Philippines: Mobile phone users 2020-2028* (2023) *Statista*. Available at: https://www.statista.com/forecasts/558756/number-of-mobile-internet-user-in-the-philippines (Accessed: December 2023).

[5] O'Neill, A. (2023) *Total population of the ASEAN countries 2018-2028*, *Statista*. Available at: https://www.statista.com/statistics/796222/total-population-of-the-asean-countries/ (Accessed: December 2023).

[6] *Financial inclusion in the Philippines - bangko sentral ng Pilipinas* (2021) *Bangko Sentral ng Pilipinas*. Available at: https://www.bsp.gov.ph/Media_And_Research/Financial%20Inclusion%20Dashboard/2021/FIDashboard_3Q2021.pdf (Accessed: December 2023).

[7] Hilario, E. (2022) *Number of unbanked Filipinos decreases as digital finance supports daily needs*, *Manila Bulletin*. Available at: https://mb.com.ph/2022/05/19/number-of-unbanked-filipinos-decreases-as-digital-finance-supports-daily-needs/ (Accessed: December 2023).

[8] *Bangko Sentral ng Pilipinas Financial Stability - directories and lists* (2023) *Bangko Sentral ng Pilipinas*. Available at: https://www.bsp.gov.ph/SitePages/FinancialStability/DirBanksFIList.aspx (Accessed: 14 December 2023).

[9] (2021) *API Dashboard*. Available at: https://apidashboard.io/apis?region=ph (Accessed: October 2023).

[10] Pymnts (2019) *Philippine fintech startup Paymongo raises $2.7m*, *PYMNTS.com - What's next in payments and commerce*. Available at: https://www.pymnts.com/news/investment-tracker/2019/philippine-fintech-startup-paymongo-funding/ (Accessed: May 2020).

[11] *What is low-code?* (no date) *IBM*. Available at: https://www.ibm.com/topics/low-code (Accessed: December 2023).

[12] *Cloud Firestore Data model | Firebase* (no date) *Google*. Available at: https://firebase.google.com/docs/firestore/data-model (Accessed: December 2023).

APPENDICES

A. Complete Program Listing

- Web Application: https://openbankph.flutterflow.app/
- Repository: https://github.com/sherbait/openbankph

B. Technical Reference

1. Hardware and Operating Systems

The web application was developed through an Edge browser (Version 120.0.2210.77), running on Windows 10 machine. Users are able to access the application using any browser, either through a desktop or mobile, and has been verified to work on Edge (Version 120.0.2210.77) and Chrome (Version 120.0.6099.110). Supported screen resolutions (px): 390x844, 1024x1366, 1440x900.

2. Programming Language

The web application was developed using low-code FlutterFlow (https://app.flutterflow.io/, v4.0) with Dart (Flutter 3.13.7). NoSQL queries to Google Firestore also use Dart.

3. Server

The application is deployed on a FlutterFlow server, using the platform.

4. Maintenance

The assets and code base are available on GitHub. The FlutterFlow project, along with its Firebase integration and API library to the Open Bank Project, can be made available upon request to the author.

Accounts to the following must be acquired:

- FlutterFlow
- Firebase with Firestore Database, Functions, Storage, Authentication
- Open Bank Project for Developers (v5.1.0)

Also see: Web Publishing - FlutterFlow Docs, Firebase Setup - FlutterFlow Docs, Getting Started | Open Bank Project

C. User Manual (App Users)

This guide aims to assist end-users of the Open Bank PH system and make the most out of the platform.

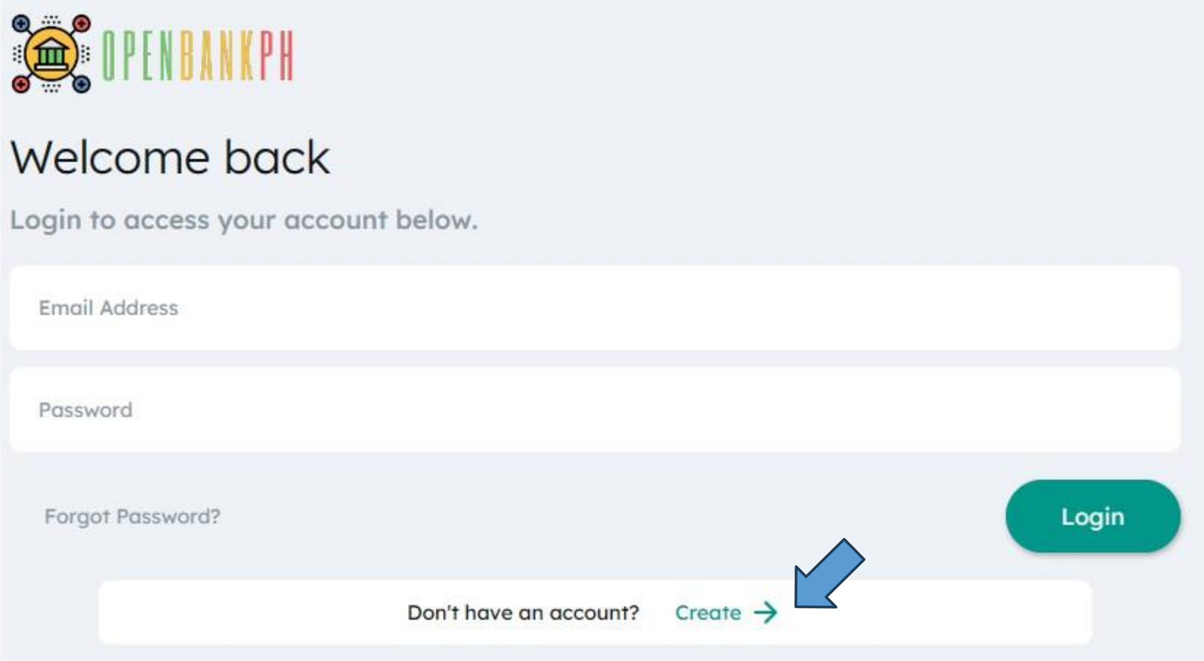
1. Getting Started

1.1. Accessing the Web Application

To access Open Bank PH, open your browser and enter the URL https://openbankph.flutterflow.app/.

1.2. Creating an Account

If you're a new user, click on the "Create" button on the login page.



Welcome back

Login to access your account below.

Don't have an account?

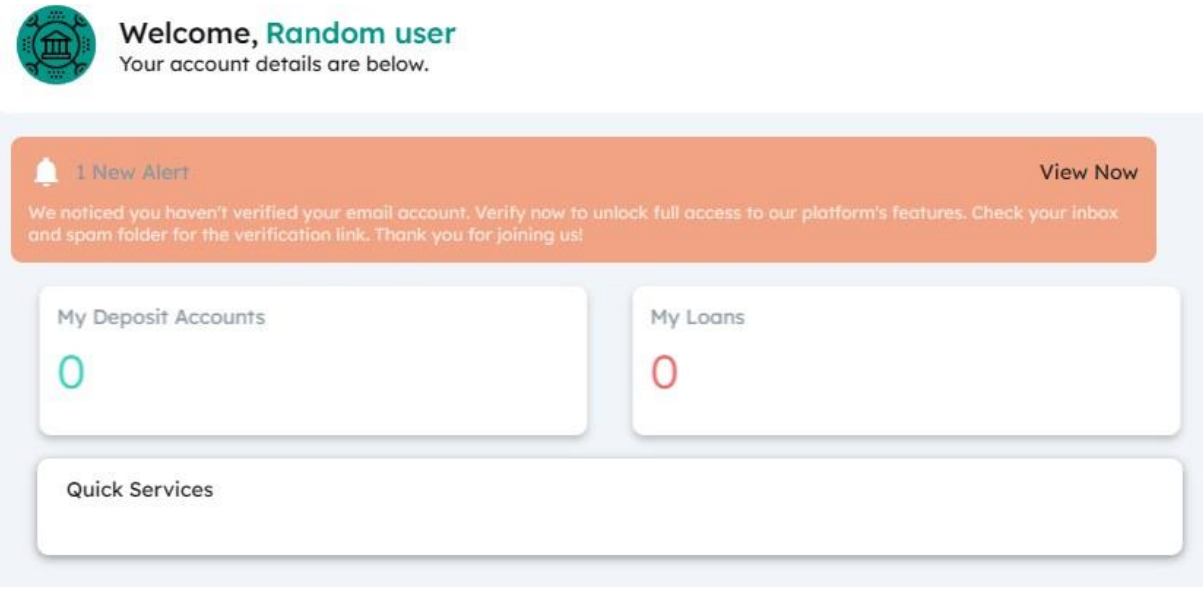
Enter the required details, such as your name, email address, and password, to create an account. After submitting, you will be redirected to another page where you can optionally customize your profile account with a picture, salutation, age, and preferred name on the application.

1.3. Logging In

For existing users, enter your registered email address and password on the login page and click "Login" to access your account.

2. Navigating the Dashboard

Upon logging in, you will be directed to the dashboard. Here, you'll find an overview of your bank accounts, navigation tabs at the bottom of the page, as well as quick links.



Welcome, Random user

Your account details are below.

1 New Alert

View Now

We noticed you haven't verified your email account. Verify now to unlock full access to our platform's features. Check your inbox and spam folder for the verification link. Thank you for joining us!

My Deposit Accounts

0

My Loans

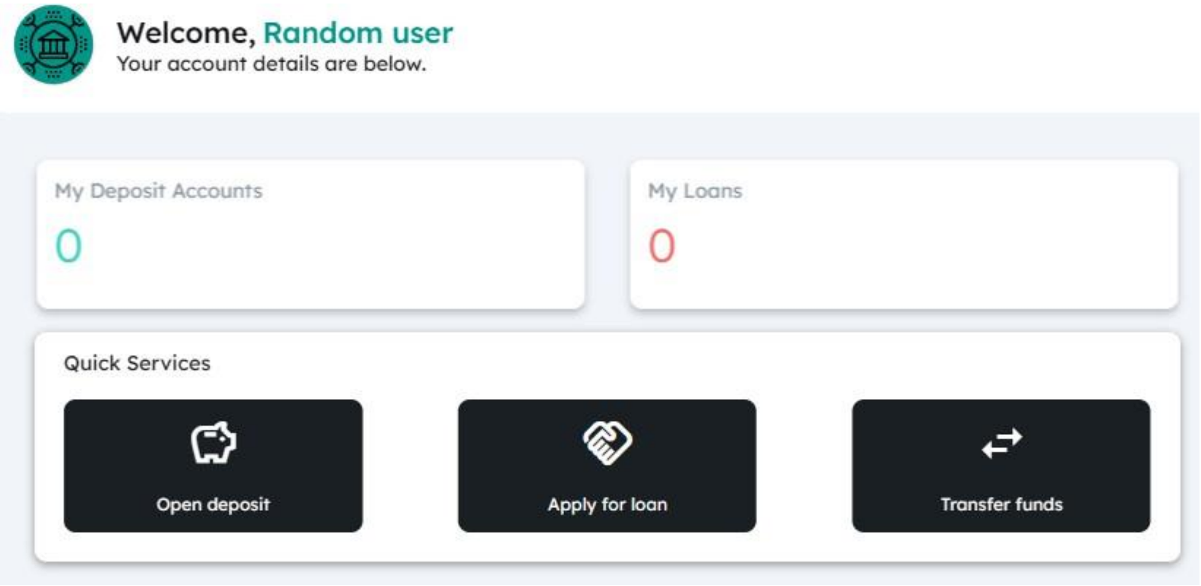
0

Quick Services

You must verify your new account to unlock other features of the platform, by clicking on the verification link sent to your email. You may also resend the verification link by clicking on the alert then "Resend verification link" button.

3. Using Core Features

Once verified, refresh the page, and your dashboard should now display more options under "Quick Services".



Welcome, Random user

Your account details are below.

My Deposit Accounts

0

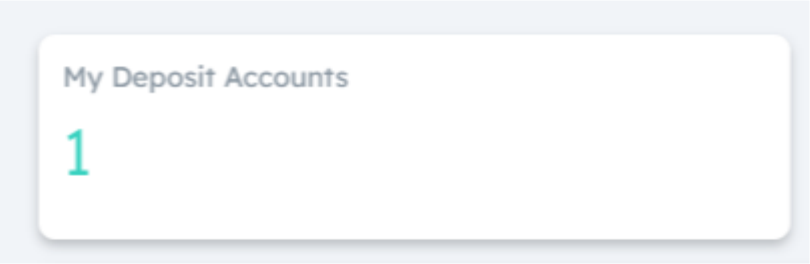
My Loans

0

Quick Services

3.1. Opening a Deposit Account

Click on the “Open deposit” button on the dashboard. Select a “Product Type” and fill in the form with your information. Submit the form and you will see a confirmation window. Click “Agree” to the terms and conditions, and you will be directed to a loading screen, and finally back to the dashboard. The number of your deposit accounts should increase by 1.

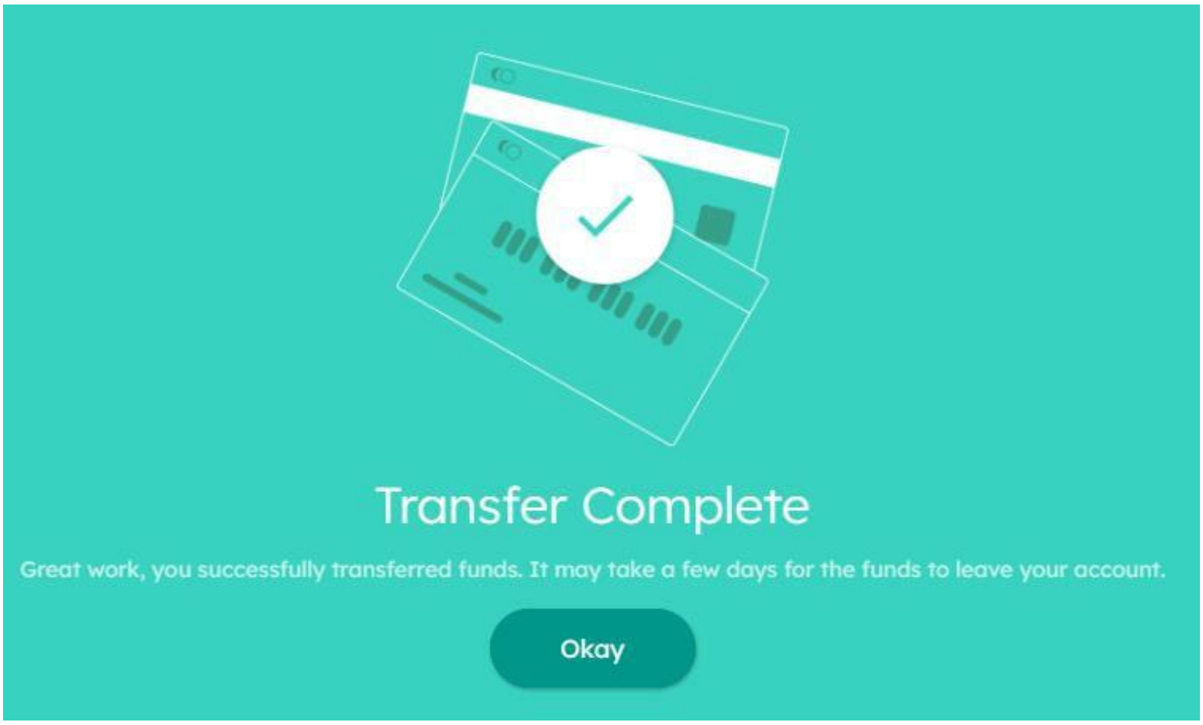


My Deposit Accounts

1

3.2. Transferring of Funds

To transfer funds, you must own at least 2 accounts. Repeat step 3.1 to create another account. With 2 accounts, click on the “Transfer funds” on the dashboard. Fill in the transfer details and click “Send Transfer” button. Once successful, you should see a successful transfer and be directed to the dashboard.

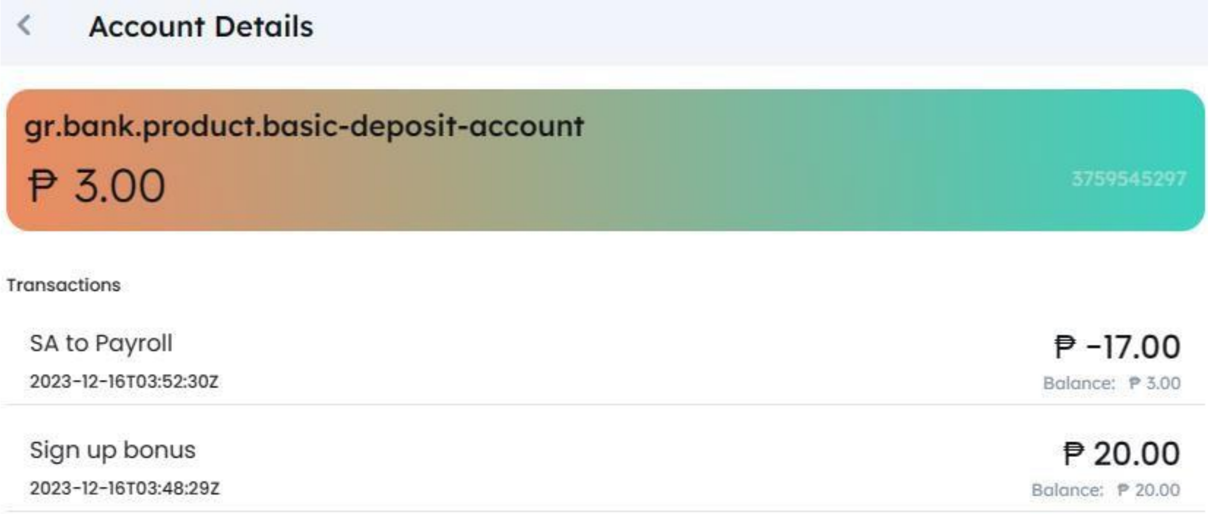


Transfer Complete

Great work, you successfully transferred funds. It may take a few days for the funds to leave your account.

3.3. Viewing Accounts and Transaction History

On the dashboard, click on the "My Deposit Accounts" widget to view a list of accounts you own. Select an account from the list and you should be directed to a page with the remaining balance of the account, account number, account name, and list of transactions.



Account Details

gr.bank.product.basic-deposit-account

₱ 3.00

3759545297

Transactions

SA to Payroll

2023-12-16T03:52:30Z

₱ -17.00

Balance: ₱ 3.00

Sign up bonus

2023-12-16T03:48:29Z

₱ 20.00

Balance: ₱ 20.00

4. Settings and Preferences

4.1. Profile Settings

To edit your profile, navigate to the profile tab at the bottom of the screen and click on "Edit Profile". You can upload a profile photo, change your display name, age, and preferred title. Click "Save Changes" to save.

4.2. Security Settings

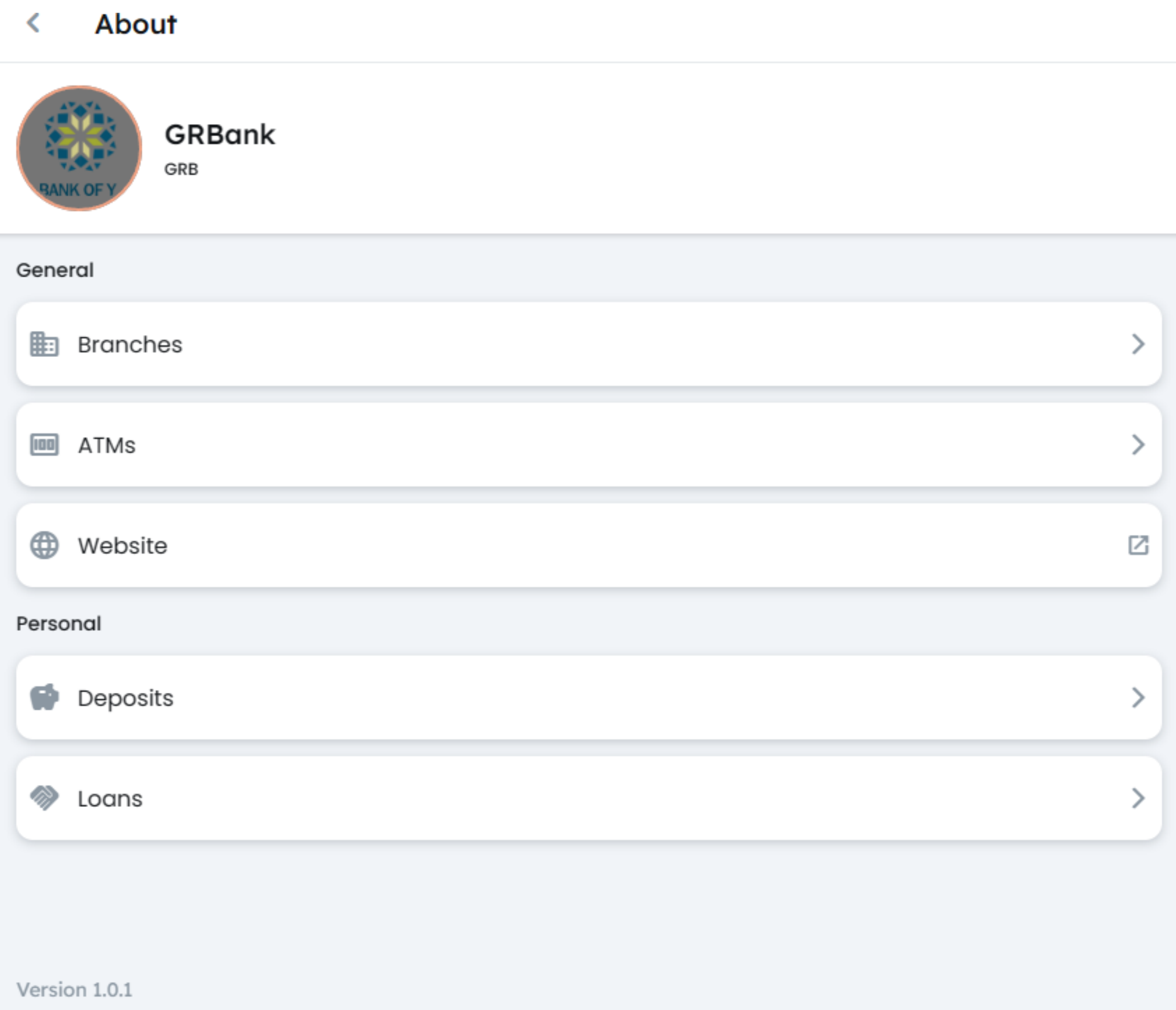
To change your password, navigate to the profile tab at the bottom of the screen and click on "Change Password". On the new screen, click "Send Reset Link". Check your email inbox with a link to change your password. Once successful, you may now log in to your account with the new password.

To recover your account, on the home page of the application, click on "Forgot Password?" On the new screen, enter your email and click "Send Reset Link" button. Check your email inbox with a link to reset your password. Once successful, you may now log in to your account with the new password.

5. Troubleshooting and Support

5.1. General Information

On the home page of the application, click on the “About Us” button. On this page, you can view the branches, ATMs, website, and product information of the bank. The current version of the application is displayed here as well.



5.2. Contacting Support

If you encounter any issues or need further assistance, log in to the application and navigate to the profile tab. Click “Support” and you will be directed to a form where you can submit a ticket. Enter your name, a short summary of the problem, and any supporting screenshots that may help your case. Click “Submit Ticket” to save.

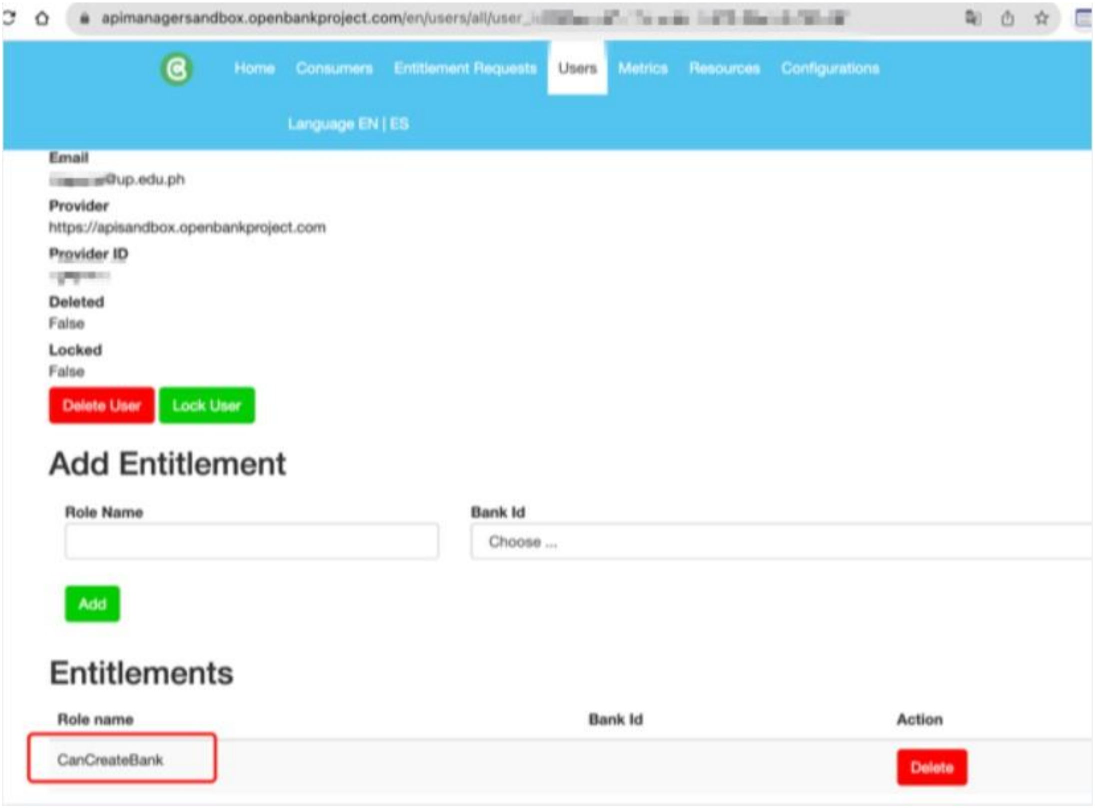
D. User Manual (Developers)

This guide aims to assist developers of the Open Bank PH system.

1. Getting Started

To replicate the environment used to develop the project, ensure you have a web browser, preferably Edge, with a stable Internet connection. Create an account on the following platforms:

- FlutterFlow. An educational license can be requested which will unlock a few more features than a standard free license, but the latter should be sufficient.
- Firebase. A free account is available, but a Blaze plan is required to access Cloud Functions, which is required to deploy to production.
- Open Bank Project. Register for free and wait for confirmation of your account within 24 hours. Follow the instructions in the linked page to test your account in the Sandbox API environment via API Explorer. When requesting your first entitlement, send a message to the developers at http://chat.openbankproject.com/ for a **CanCreateBank** role. Verify that you have the role by going to the API Manager and logging in with your registered account. Navigate to the “Users” tab and you should see the role under “Entitlements”.



Email
[redacted]@up.edu.ph

Provider
https://apisandbox.openbankproject.com

Provider ID
[redacted]

Deleted
False

Locked
False

Add Entitlement

Role Name:

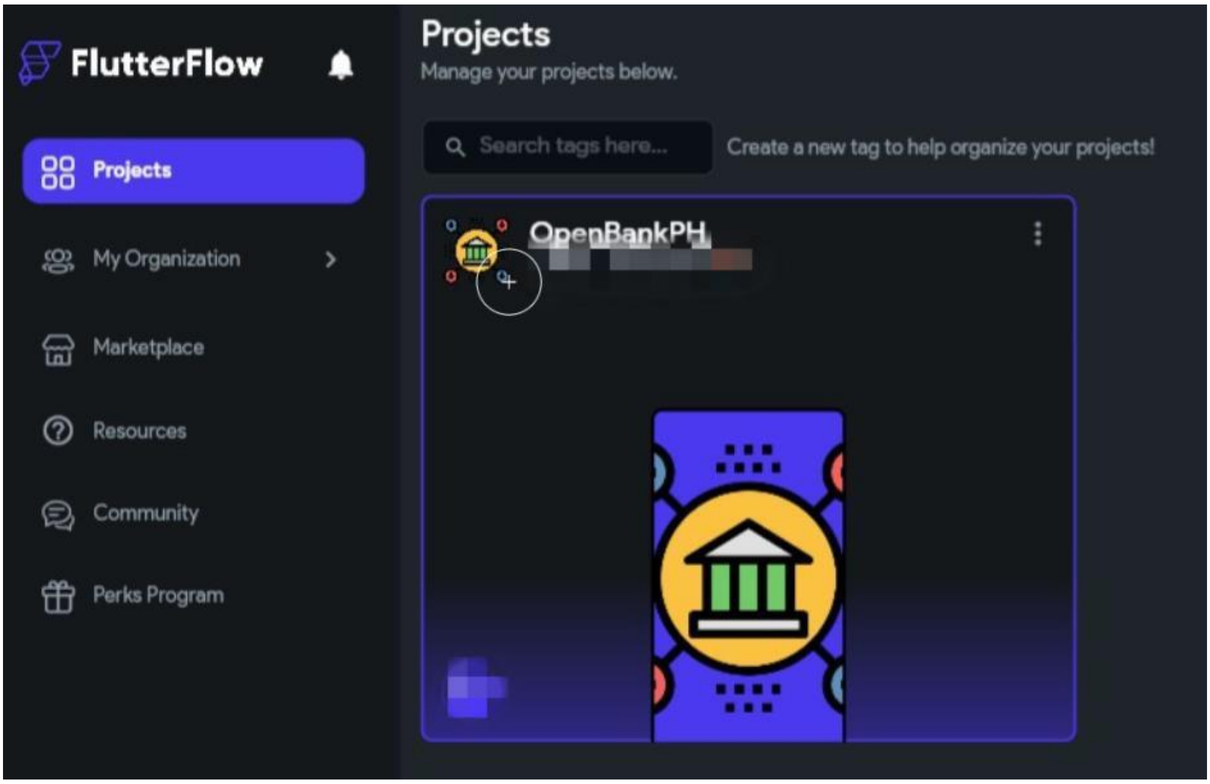
Bank Id: Choose ...

Entitlements

Role name	Bank Id	Action
CanCreateBank		Delete

2. Integration and Usage Guides

Main development is done through FlutterFlow's developer dashboard. You can create a new project then set up Firebase with it, or request the author for access to the OpenBankPH project itself.



Projects

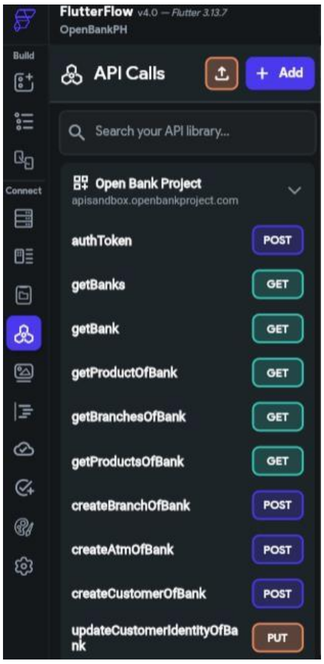
Manage your projects below.

Create a new tag to help organize your projects!

[redacted]

Alternatively, you can check out the GitHub code and build the application by [setting up FlutterFlow on your machine](). There is currently no way to import a GitHub code to auto-generate a FlutterFlow project through the dashboard UI.

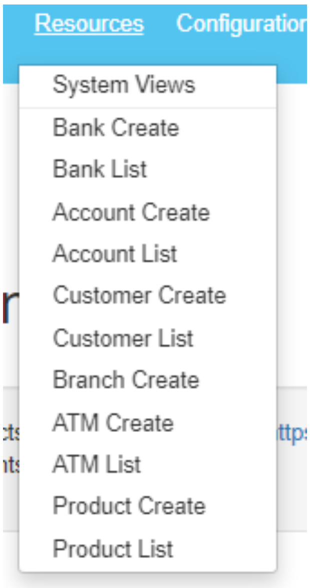
After setting up your FlutterFlow project, review the API library tab, where you can build your library of API calls to be used by the application. In the OpenBankPH project, OBP endpoints are stored here.



When using certain OBP endpoints, you may encounter an authorization error. You may can set up authentication in different ways:

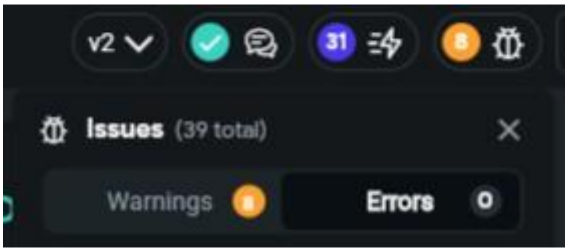
- Direct Login
- OAuth 1.0 / 2.0
- Gateway Login

For API requests that return with a role error, go to the API Manager and assign the missing entitlement to your account. The API Manager also has admin interfaces to create entities without using the API Explorer.



3. Testing and Debugging

FlutterFlow is able to identify errors or warnings as you develop the app through the developer dashboard.



You can usually ignore Warnings but not Errors because it will prevent you from deploying to the Test Environment. For more information on how to access this environment, see Testing Your App - FlutterFlow Docs.

4. Support and Resources

FlutterFlow has a comprehensive documentation with samples and videos at https://docs.flutterflow.io/.

For specific queries or technical support, contact the author at ddgepte@up.edu.ph.