Republic of the Philippines
BATANGAS STATE UNIVERSITY
The National Engineering University
ARASOF-Nasugbu Campus
R. Martinez St., Brgy. Bucana, Nasugbu, Batangas, Philippines 4231
Tel Nos.: (+63 43) 416-0350
E-mail Address: cics.nasugbu@g.batstate-u.edu.ph | Website Address: http://www.batstate-u.edu.ph

College of Informatics and Computing Sciences

Q.ATool Bank
Online Bank Withdrawal

A Final Project
Presented to the Course Specialists of the
College of Informatics and Computing Sciences of
Batangas State University ARASOF Nasugbu
Nasugbu, Batangas

In Partial Fulfilment of the Requirements for
IT 414 System Quality Assurance

by

Andino, Cristie
Andrin, Erlyn
Bautista, Joy-Lyn
Gabriel, Jhorely
Marasigan, Carl Joriz Mickeal

October 2023

LOGO

I. SYSTEM REQUIREMENTS

1. Security Requirements

- The user must enter his/her username, password, and OTP to access the system.
- All the functions are available only to the user who logged in to the account.
- The user is the only authorized and capable person to withdraw money from the account.
- The system must display a message prompt when the withdrawal is successfully sent, given a 6-digit pin sent to your mobile number for verifying when you claim the money.

2. Overall Requirements

- The online banking system should do withdrawal process.
- The withdrawal process must be automatically updated the balance real time.
- The system must be user friendly, having clean and simple interfaces.

II. ANALYSIS

Services	Components	Costs
Cloud Computing Services	Transaction Model: Create a model for recording withdrawal transactions, including attributes like amount, date, and account association. One-Time PIN Model: Specify attributes and logic for generating, delivering, and validating one-time PINs.	400 Dollars or 22,789.20 Pesos

	User Authentication: Implement logic for user authentication through the mobile banking app based on the user model. Transaction Processing: Develop logic to handle withdrawal requests, update account balances, and record transactions in the transaction model. ATM Model: Describe ATM attributes, authentication process using one-time PINs, and transaction processing.	
SMS Services	Transaction Initiation: Develop logic for users to initiate cardless withdrawal transactions using the app.	130 Dollars or 7,348.95 Pesos
Cybersecurity/Security Services	Security Model: Incorporate a model for encryption, secure data transmission, and access control. Security Measures: Integrate encryption, secure data transmission protocols, and access control mechanisms.	2,000 Dollars or 113,946.00 Pesos

III. GANTT CHART

PROJECT PLAN TEMPLATE

PROJECT TITLE: Online Bank Withdrawal

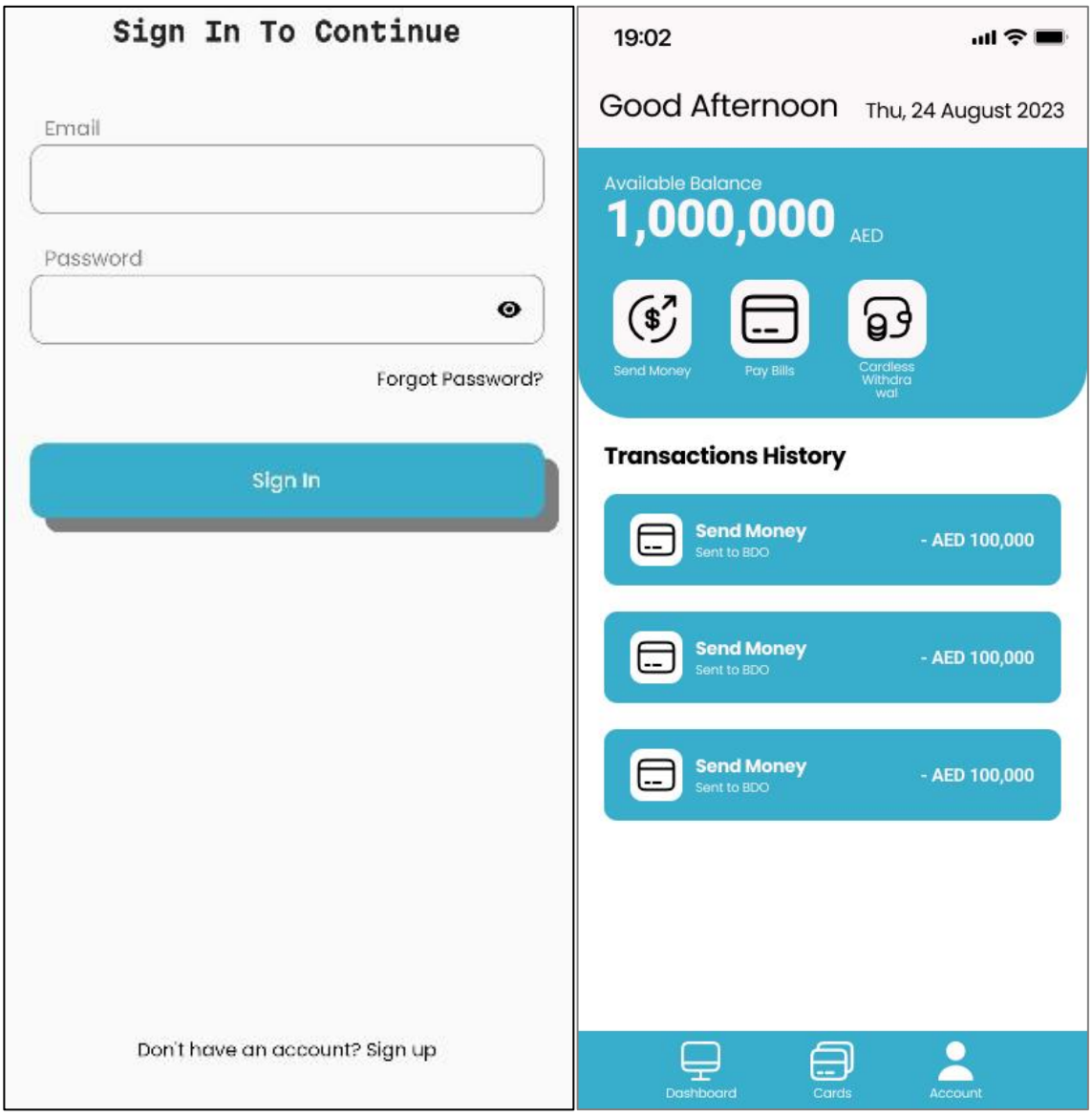
START DATE: 08/23

END DATE: 11/27

PROJECT DURATION in days: 97

WBS NO.	TASK NAME	STATUS	ASSIGNED TO	START DATE	END DATE	DURATION in days	COMMENTS
1.1	**INITIATION**	**Complete**	**Everyone**	08/23	08/29	**7**	
1.1.1	– Project Kickoff Meeting	Complete	Everyone	08/28	08/28	1	
1.1.2	– Develop the Propose Design	Complete	Everyone	08/23	08/29	7	
1.2	**PLANNING**	**Complete**	**Everyone**	08/23	09/26	**35**	
1.2.1	– Project Group Meeting	Complete	Everyone	09/26	09/26	1	
1.2.2	– Identify Business Process	Complete	Everyone	08/23	08/29	7	
1.2.3	– Create UI/UX Designs	Complete	Carl Joriz Mickeal Marasigan, Jhorely	08/23	08/29	7	
1.2.4	– Identify Programming Languages to be Used	Complete	Carl Joriz Mickeal Marasigan	08/30	08/31	2	
1.2.5	– Create and Validate the Database Schema	Complete	Carl Joriz Mickeal Marasigan, Jhorely	09/01	09/15	15	
1.3	**EXECUTION**	**In Progress**	**Everyone**	08/23	11/24	**94**	
1.3.1	– Project Group Meeting	Complete	Everyone	10/01	10/01	1	
1.3.2	– Develop a Database	In Progress	Carl Joriz Mickeal Marasigan, Jhorely	09/01	11/24	85	
1.3.3	– Develop a Mobile App	In Progress	Carl Joriz Mickeal Marasigan, Jhorely	09/01	11/24	85	
1.4	**CONTROL**	**In Progress**	**Everyone**	08/23	11/27	**97**	
1.4.1	– Project Group Meeting	Not Started	Everyone	11/07	11/07	1	
1.4.2	– Project Quality Check	In Progress	Erlyn Andrin, Cristie Andino	09/01	11/24	85	
1.4.3	– Project Testing/security	In Progress	Joy-lyn Bautista	09/01	11/24	85	
1.4.4	– Review of Business Process	In Progress	Everyone	09/25	11/24	61	
1.4.5	– Deploy live	In Progress	Carl Joriz Mickeal Marasigan	09/25	11/24	61	
1.4.6	– Update Gantt Chart	In Progress	Everyone	08/23	11/27	97	
1.5	**CLOSEOUT**	**Not Started**	**Everyone**	08/23	11/27	**97**	
1.5.1	– Project Group Meeting	Not Started	Everyone	11/21	11/21	1	
1.5.2	– Document Lessons Learned	Not Started	Erlyn Andrin, Cristie Andino, Joy-lyn	11/27	11/27	1	
1.5.3	– Pre Oral	Not Started	Everyone	10/04	10/06	3	
1.5.4	– Project Pitching	Not Started	Everyone	11/26	11/26	1	
1.5.5	– Archive Files/Documents	Not Started	Everyone	11/27	11/27	1	

IV. DESIGN


Sign In To Continue
Email
Password
Forgot Password?
Sign In
Don't have an account? Sign up
19:02
Good Afternoon
Thu, 24 August 2023
Available Balance
1,000,000
AED
Send Money
Pay Bills
Cardless Withdra wal
Transactions History
Send Money
Sent to BDO
- AED 100,000
Send Money
Sent to BDO
- AED 100,000
Send Money
Sent to BDO
- AED 100,000
Dashboard
Cards
Account

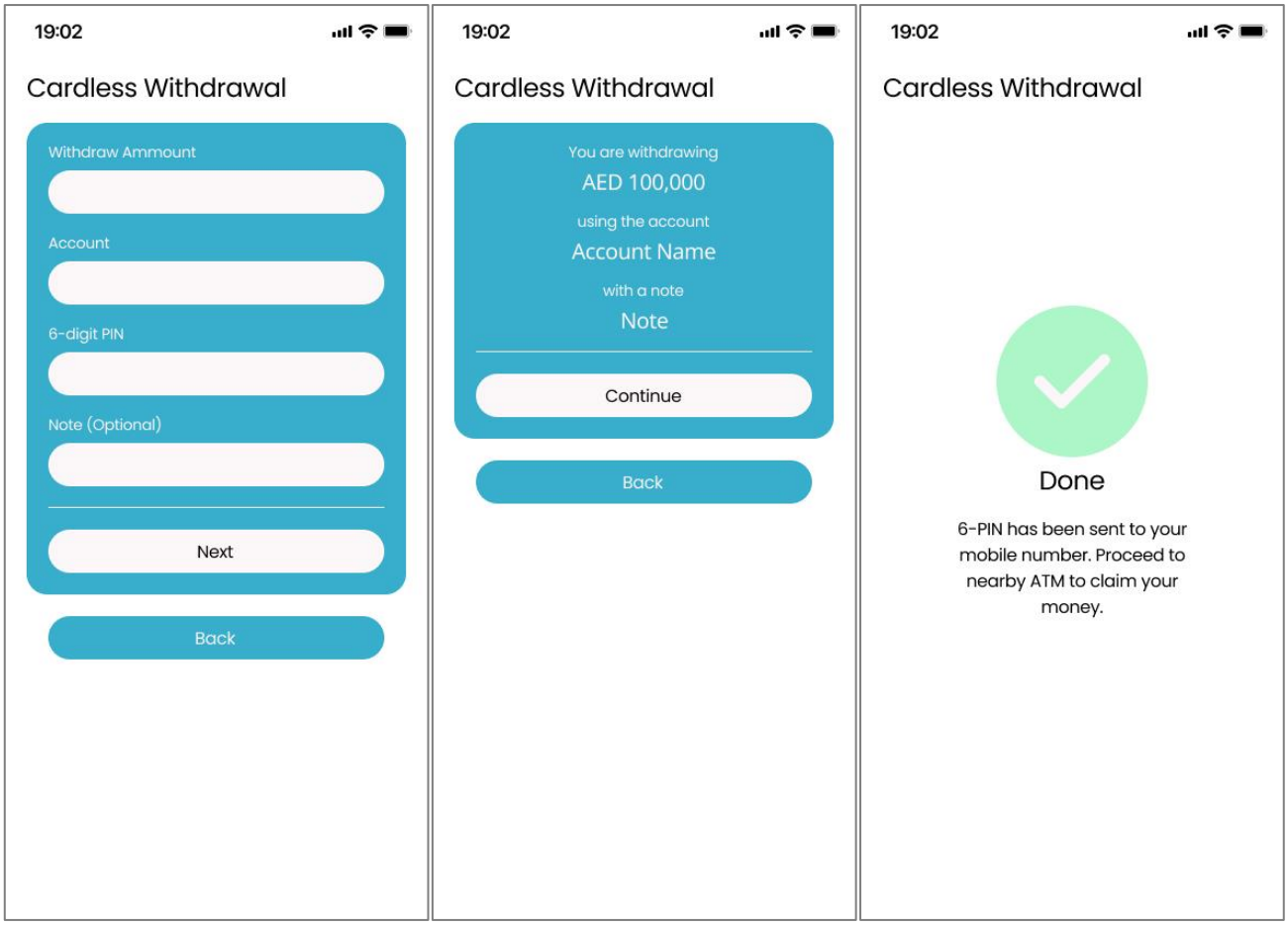


Figure 4.1 System Interfaces for Mobile Application

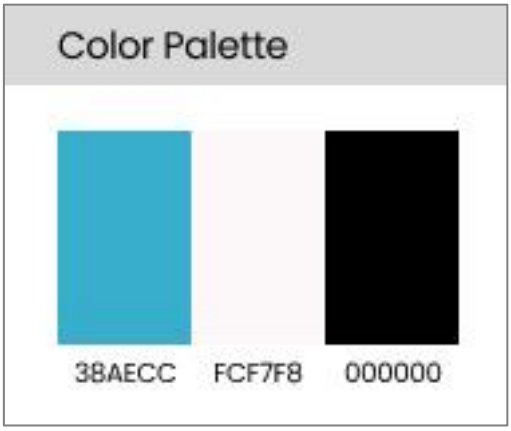


Figure 4.2 Color Palette

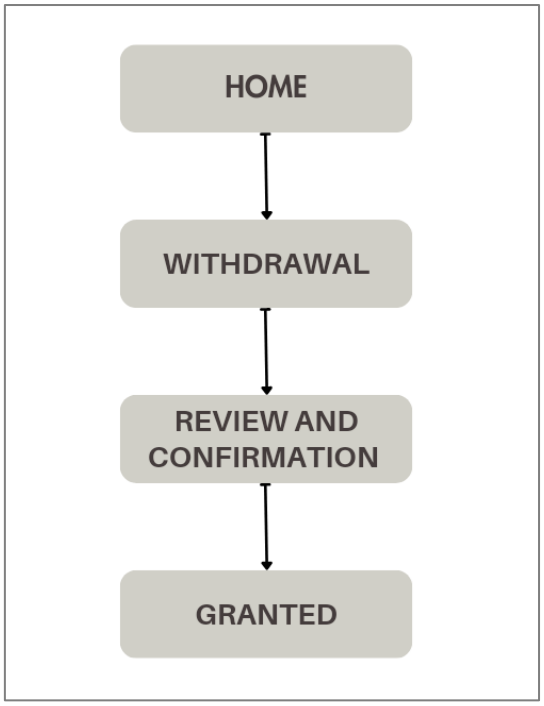


Figure 4.3 System Flow

Software	**Specification**	**Function**
IDE	Visual Studio Code	It enables developers to write HTML, CSS, JavaScript, and PHP codes for the web system.
IDE	Android Studio	It enables developers to write dart codes for mobile application.
Web Server Solution	XAMPP	It enables developers to test the system in a test environment. Apache and MySQL are included in XAMPP.
Markup Language	HTML	HTML is used to describe the structure of the web page and it consists of series of elements.
Style Sheet	CSS	CSS describes the styles or how the HTML elements should appear on the screen.
Programming Language	PHP	PHP is a scripting language embedded into HTML.
Database	MySQL	MySQL is a relational database management system used as a data repository.
Programming Language	Dart	Dart is a programming language used to develop mobile application.

Table 4.1 Software needed for development

V. CODING

A. confirmation.dart

```
import 'package:flutter/material.dart';
import 'package:qatool/Done.dart';
import 'package:qatool/withdrawal.dart';
import 'package:qatool/withdrawal_input.dart';

class ConfirmationPage extends StatefulWidget {
 const ConfirmationPage({super.key});

 @override
 State<ConfirmationPage> createState() => _ConfirmationPageState();
}

class _ConfirmationPageState extends State<ConfirmationPage> {
 int? amount;
 String? note;

 @override
 void initState() {
  amount = WithdrawalInput().getAmount;
  note = WithdrawalInput().getNote;
  super.initState();
 }
 @override
 Widget build(BuildContext context) {
  return Scaffold(
   body: Center(
    child: Padding(
     padding: const EdgeInsets.fromLTRB(20.0, 50.0, 20.0, 20.0),
     child: Column(
      children: [
       /*** HEADER ***/
       const Align(
        alignment: Alignment.centerLeft,
        child: Text(
         "Cardless Withdrawal",
         style: TextStyle(
            fontFamily: "Poppins-Medium",
            fontSize: 23.0
         ),
        ),
       ),

       const SizedBox(height: 20.0),
```

B. done.dart

```
import 'package:flutter/material.dart';
import 'home.dart';

class DonePage extends StatefulWidget {
  const DonePage({super.key});

  @override
  State<DonePage> createState() => _DonePageState();
}

class _DonePageState extends State<DonePage> {


  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Padding(
          padding: const EdgeInsets.fromLTRB(20.0, 50.0, 20.0, 20.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [
              /*** HEADER ***/
              const Align(
                alignment: Alignment.centerLeft,
                child: Text(
                  "Cardless Withdrawal",
                  style: TextStyle(
                      fontFamily: "Poppins-Medium",
                      fontSize: 23.0
                  ),
                ),
              ),

              /*** MAIN CONTENT ***/
              const Column(
                children: [
                  Icon(
                    Icons.check_circle,
                    color: Color.fromRGBO(129, 255, 91, 1),
                    size: 200.0,
                  ),

                  Text(
```

C. home.dart

```
import 'package:flutter/material.dart';
import 'package:qatool/user_details.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:add_comma/add_comma.dart';
import 'package:http/http.dart' as http;
import 'package:qatool/withdrawal.dart';

class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  String? id;
  late String balance;
  bool isSet = false;

  Future<void> getId () async {
    final SharedPreferences pref = await SharedPreferences.getInstance();
    id = pref.getString('userID');

    UserDetails().setID(id!);
    setBalance();
  }

  Future<void> setBalance() async {
    var url = Uri.http(
        "adminvadd.me",
        "bank/getuserbalance.php",
        {
          "details": "balance",
          "id": id
        }
    );
    var response = await http.get(url);

    balance = response.body;

    final putComma = addCommas();
    UserDetails().setBalance(int.parse(balance));
    balance = putComma(int.parse(balance))!;
    isSet = true;
```

D. main.dart

```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'package:shared_preferences/shared_preferences.dart';
import 'home.dart';

void main() {
  runApp(const MaterialApp(
    debugShowCheckedModeBanner: false,
    home: LoginPage(),
  ));
}

class LoginPage extends StatefulWidget {
  const LoginPage({super.key});

  @override
  State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  bool hide = true;

  TextEditingController email = TextEditingController();
  TextEditingController password = TextEditingController();

  Future<void> setId() async {
    var url = Uri.http(
      "adminvadd.me",
      "bank/getuserdetails.php",
      {
        "details": "id",
        "email": email.text,
        "password": password.text
      }
    );
    var response = await http.get(url);

    String id = response.body;

    final SharedPreferences pref = await SharedPreferences.getInstance();
    pref.setString('userID', id);
  }

  Future<bool> verifyAuth() async {
```

E. user_details.dart

```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'package:shared_preferences/shared_preferences.dart';
import 'home.dart';

void main() {
  runApp(const MaterialApp(
    debugShowCheckedModeBanner: false,
    home: LoginPage(),
  ));
}

class LoginPage extends StatefulWidget {
  const LoginPage({super.key});

  @override
  State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  bool hide = true;

  TextEditingController email = TextEditingController();
  TextEditingController password = TextEditingController();

  Future<void> setId() async {
    var url = Uri.http(
      "adminvadd.me",
      "bank/getuserdetails.php",
      {
        "details": "id",
        "email": email.text,
        "password": password.text
      }
    );
    var response = await http.get(url);

    String id = response.body;

    final SharedPreferences pref = await SharedPreferences.getInstance();
    pref.setString('userID', id);
  }

  Future<bool> verifyAuth() async {
```

F. withdrawal.dart

```
import 'package:flutter/material.dart';
import 'package:qatool/confirmation.dart';
import 'package:qatool/home.dart';
import 'package:qatool/user_details.dart';
import 'package:http/http.dart' as http;
import 'withdrawal_input.dart';

class WithdrawalPage extends StatefulWidget {
  const WithdrawalPage({super.key});

  @override
  State<WithdrawalPage> createState() => _WithdrawalPageState();
}

class _WithdrawalPageState extends State<WithdrawalPage> {
  TextEditingController amount = TextEditingController();
  TextEditingController pin = TextEditingController();
  TextEditingController note = TextEditingController();
  int balance = UserDetails().getBalance;

  bool checkInput() {
    bool amountEmtpy = false;
    bool pinEmpty = false;
    if(amount.text.isEmpty) {
      amountEmtpy = true;
    }
    if(pin.text.isEmpty || pin.text.length != 6) {
      pinEmpty = true;
    }
    if(amountEmtpy || pinEmpty) {
      return true;
    }
    return false;
  }

  bool checkAmount() {
    int inputAmount = int.parse(amount.text);
    if(inputAmount > balance) {
      return true;
    }
    return false;
  }

  Future<bool> checkPin() async {
```

G. withdrawal_input.dart

```
class WithdrawalInput {
  static int? amount;
  static String? note;

  get getAmount {
    return amount!;
  }

  get getNote {
    return note;
  }

  void setAmount(int value) {
    amount = value;
  }

  void setNote(String value) {
    note = value;
  }
}
```

VI. TESTING

Test Case Type	Description	Test Step	Expected Result	Status
Functionality	Test the functionality of the login page	Open the application. Navigate to the online banking portal. Enter valid login credentials (username and password). Click the "Log In" button.	The user should be successfully logged in and redirected to the dashboard.	Pass or Fail
Usability	Ensure the cardless withdrawal option is accessible	Locate and click on the "Cardless Withdrawal" option. Enter the withdrawal amount. Provide any required additional information, such as account, 6-digit PIN and note (optional). Click the "Next" button.	The system should accept the request and proceed to the next step.	Pass or Fail
Functionality	Ensure that the details inputted is accurate	Review the withdrawal details displayed on the confirmation screen. Ensure that the withdrawal amount and recipient information (if applicable) are accurate.	The system should display a confirmation message indicating that the cardless withdrawal request has been received.	Pass or Fail

		Confirm the transaction by clicking the "Continue" button.		
Functionality	Ensure that the system sends an OTP to users	Check for an SMS confirmation regarding the 6-digit PIN or OTP. Verify that the confirmation contains the correct transaction details.	The user should receive a confirmation notification with accurate transaction information.	Pass or Fail
Usability	Ensure that the OTP sent is working by proceeding to any ATM kiosk	Go to any ATM location as indicated during the cardless withdrawal request. Follow the on-screen instructions to complete the withdrawal, which may include entering a reference number or a one-time code received through SMS.	The ATM should dispense the requested amount of cash, and the transaction should be successful.	Pass or Fail

Table 6.1 Test Cases

Step #	Description	Expected Result	Test Result	Comments
1	Users need to register for the cardless access system. They provide their personal information, including email and create a secure password.	User data is stored securely in the database, and a confirmation email or SMS is sent to the user.	User registration is successful. Data is stored in the database, and the confirmation message is sent.	Ensure that user data is encrypted and stored securely to protect sensitive information.
2	Users visit the login page and enter their credentials (email and password).	Users are authenticated, and they are redirected to the cardless withdrawal page.	User authentication is successful, and users are redirected to the cardless withdrawal page upon entering valid credentials.	Implement strong password policies and account lockout mechanisms to enhance security.
3	Users arrive at the cardless withdrawal page, where they input the withdrawal amount.	Users should be able to input the amount they want to withdraw.	Users can input the withdrawal amount, account, 6 digit PIN	Validate input fields to prevent incorrect or malicious data entry.
4	Users review the withdrawal details and confirm the transaction	Users see a summary of the transaction and can confirm it by clicking a "Continue" button	Users can review the transaction details and successfully confirm the withdrawal.	Include transaction details such as withdrawal amount, destination account, and transaction fee, if applicable.
5	After confirmation, users receive an OTP (One-Time Password) on their registered mobile number. They need to enter this OTP to authorize the transaction	Users receive an OTP and can enter it to authorize the withdrawal.	Users receive the OTP and can enter it to authorize the transaction.	Ensure OTPs are time-sensitive and can only be used once.
6		The ATM should dispense the requested amount of cash, and the transaction should be successful.	Go to the specified ATM location as indicated during the cardless withdrawal request. Follow the on-screen instructions to complete the withdrawal, which	Ensure that the ATM successfully dispenses the requested amount of cash and completes the transaction. The user should follow on-screen

			may include entering a reference number or a one-time code received through SMS.	instructions, which may include entering a reference number or a one-time code received through SMS. This step should be straightforward and user-friendly to minimize errors and provide a smooth user experience.

Table 6.2 Test Scripts

VII. INSTALLATION AND CONVERSION

To access the system's interface, users must meet specific software requirements. These include having an Android device running version 11 or above, equipped with a minimum of 2GB of RAM, and ensuring an active internet connection. Additionally, it's essential that the user's SIM card maintains a signal strength to receive the One-Time Password (OTP) required for login authentication. These prerequisites ensure a seamless and secure login experience for users accessing the system.