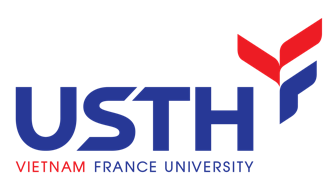
**UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HA NOI**



**BACHELOR THESIS**

Academic year: 2020 - 2023

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project: **Building a SmartOTP solution for**

**electronic transactions.**

Deployment and application on SmartEVN application of EVN

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Author: Nguyen Hoai Phuong

Student ID: BA10-046

Internal Supervisor: Dr. ***Doan Nhat Quang***

External Supervisor: Mr. ***Doan Van Long***

Host company: Electricity of Viet Nam- NCSC

**Ha Noi, July 20**

# ATTESTATION

I, Nguyen Hoai Phuong, confirm that my report is free from any form of plagiarism, including copying and pasting from other sources, without providing proper citations.

I am fully aware of the repercussions associated with plagiarism, and I understand that if any instances of plagiarism are detected, both my thesis and all the results of my internship will be rejected. In such a scenario, I acknowledge my full responsibility and willingly accept any appropriate penalties imposed by the board and the university.

Tôi, Nguyễn Hoài Phương, xin ký tên dưới đây nhằm cam kết rằng bài báo cáo của tôi không đạo văn (cố ý sao chép) mà không có trích dẫn đầy đủ.

Nếu cố tình sai phạm, tôi hoàn toàn nhận thức được hậu quả và hiểu rằng bài báo cáo cùng toàn bộ kết quả thực tập của tôi sẽ không được chấp nhận. Trong trường hợp đó, tôi xin nhận toàn bộ trách nhiệm cũng như hình thức kỷ luật tương xứng của phía hội đồng bảo vệ và nhà trường

Ngày 02 tháng 07 năm 2023

Chữ ký/ Signature

**Nguyễn Hoài Phương**

# ACKNOWLEDGMENTS

This project is a product by Nguyen Hoai Phuong at USTH Vietnam France University under the support of my supervisor **Mr.Doan Van Long** and **Dr.Doan Nhat Quang**

First, I would like to express my appreciation to **Mr.Doan Van Long**, my external supervisor from the Electricity of Viet Nam (EVN), for his dedicated guidance and invaluable support throughout my internship. Mr. Long played a vital role in enhancing my technical knowledge and honing my skills as a member of his team. His exceptional leadership and expertise in team management were instrumental in my professional growth.

I am also immensely grateful to **Dr. Doan Nhat Quang**, my internal supervisor from the **University of Science and Technology of Hanoi (USTH). Dr. Quang's** proactive approach and valuable suggestions greatly contributed to the success of my project. His continuous support and assistance ensured that I had the necessary resources and guidance to excel in my work.

Furthermore, I would like to extend my sincere gratitude to both **USTH** and **EVN** for providing me with the opportunity to work on this project. Their trust and confidence in my abilities have been immensely rewarding, and I am grateful for the invaluable experience and knowledge gained during this internship.

I am truly privileged to have had such dedicated mentors and organizations that have supported me throughout this journey.

Hanoi, July 02th, 2023

Nguyen Hoai Phuong

# ABSTRACT

This report discusses the development of a secure electricity bill payment application using the Smart OTP (Two-Factor Authentication) (One-Time Password) service. In an era where digital transactions between bank accounts play a crucial role, protecting sensitive financial information becomes extremely important. The main goal of this application is to support easy and secure electricity bill payments by integrating the Smart OTP system.

The report describes the process of designing and implementing the payment platform, including a user-friendly interface, real-time bill retrieval, and instant payment processing. Additionally, it explores the integration of the Smart OTP service, adding an extra layer of security through two-factor authentication, which enhances user trust in the application.

Throughout the development process, various programming languages, technologies, and security protocols were used to ensure the system's reliability and integrity. The report also discusses the lessons learned and insights gained from this project, such as the importance of following standards and teamwork, as well as the complexities of incorporating advanced security measures.

In conclusion, this project not only presents an effective solution for electricity bill payments but also highlights the significance of strong security measures in today's digital financial landscape. The integration of the Smart OTP service serves as evidence of our commitment to protecting user data and security.

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| OTP | One time Password |
| TOTP | Time base-on One time password |
| HOTP | HMAC-Base-on One time password |
| RSA | Rivest, Shamir, Adleman |
| JWT | Json Web Token |
| UC | Use Case |
| HMAC | Hash-Based Message Authentication Codes |
| SRS | Software Requirements Specification |
| API | Application Programming Interface |
| DB | Database |
| SMS | Short Message Services |

# LIST OF FIGURES

[Figure 1: Cyber Threat Landscape 11](#_heading=h.17dp8vu)

[Figure 2: Client View Map 11](#_heading=h.26in1rg)

[Figure 3: Define States 12](#_heading=h.35nkun2)

[Figure 4: Member Division Role Overview 14](#_heading=h.4i7ojhp)

[Figure 5: Module Core Map 17](#_heading=h.49x2ik5)

[Figure 6: Use Case Diagram 18](#_heading=h.3o7alnk)

[Figure 7: System Architecture 28](#_heading=h.46r0co2)

[Figure 8: WAF Architecture 29](#_heading=h.111kx3o)

[Figure 9: Cloudflare DNS Interface 30](#_heading=h.3l18frh)

[Figure 10: Loading & Login User Interface 35](#_heading=h.1rvwp1q)

[Figure 11: Dashboard Interface 35](#_heading=h.4bvk7pj)

[Figure 12: Uptime Overview Interface 36](#_heading=h.2r0uhxc)

[Figure 13: Uptime Detail Interface 37](#_heading=h.1664s55)

# LIST OF TABLES

[Table 1: Compare Between 3 Tools Table 13](#_heading=h.2jxsxqh)

[Table 2: Tools & Techniques 29](#_heading=h.2lwamvv)

[Table 3: Compare between alpha version and beta version. 32](#_heading=h.2zbgiuw)

# TABLE OF CONTENT

[ACKNOWLEDGMENTS 2](#_Toc142953423)

[ABSTRACT 3](#_Toc142953424)

[LIST OF ABBREVIATIONS 4](#_Toc142953425)

[LIST OF FIGURES 5](#_Toc142953426)

[LIST OF TABLES 5](#_Toc142953427)

[TABLE OF CONTENT 6](#_Toc142953428)

[Chapter 01: Introduction 8](#_Toc142953429)

[1. Problem Statement and Project Statement 8](#_Toc142953430)

[2. Context and Motivation? 8](#_Toc142953431)

[2.1 Context. 8](#_Toc142953432)

[2.2 Motivation 9](#_Toc142953433)

[3. Compare Smart OTP and SMS OTP 10](#_Toc142953434)

[Chapter 02: Background 11](#_Toc142953435)

[1. Expected Outcome 11](#_Toc142953436)

[2. Overall System 11](#_Toc142953437)

[3. Expected Features 13](#_Toc142953438)

[4. Executed Features 13](#_Toc142953439)

[5. Two factor authentication service 15](#_Toc142953440)

[5.1. System does not use two-authentication factor. 15](#_Toc142953441)

[5.2. System uses two-authentication factor. 15](#_Toc142953442)

[6.Smart OTP works 16](#_Toc142953443)

[6.1. Register Smart OTP 16](#_Toc142953444)

[6.2. Smart OTP verification 16](#_Toc142953445)

[6.3. OTP code generation 17](#_Toc142953446)

[6.4 Security capabilities of the system 20](#_Toc142953447)

[Chapter 03: Analysis and Design 23](#_Toc142953448)

[1. System Architecture 23](#_Toc142953449)

[2. Use Cases 24](#_Toc142953450)

[3. User Characteristics 25](#_Toc142953451)

[4. Use Cases and Scenario Description 25](#_Toc142953452)

[4.1. Module: Register and Login 26](#_Toc142953453)

[4.2. Module: Register Smart OTP 26](#_Toc142953454)

[4.3. Module: Payment 26](#_Toc142953455)

[4.4. Module: Verify OTP 26](#_Toc142953456)

[4.5. Expected Features 26](#_Toc142953457)

[4.6. Executed Features 26](#_Toc142953458)

[5. Sequence Diagram 26](#_Toc142953459)

[5.1. System Authentication 26](#_Toc142953460)

[5.2. Check bill 26](#_Toc142953461)

[5.3. Sign up Smart OTP 26](#_Toc142953462)

[5.4. OTP Verification 26](#_Toc142953463)

[5.5. Payment 26](#_Toc142953464)

[6. Functional requirements 26](#_Toc142953465)

[7. Non-functional requirements 26](#_Toc142953466)

[Chapter 04: Material and Method 27](#_Toc142953467)

[1. System Architecture and Technical Choice 27](#_Toc142953468)

[Chapter 05: Results 30](#_Toc142953469)

[1. Results 30](#_Toc142953470)

[2. Product Preview 31](#_Toc142953471)

[Chapter 06: Conclusion and Future Work 32](#_Toc142953472)

[1. Conclusion 32](#_Toc142953473)

[2. Future Work 32](#_Toc142953474)

[References 33](#_Toc142953475)

[Annex 34](#_Toc142953476)

# Section 01: Introduction

*Introduce the context and the problem that we are trying to solve & also the project’s objective regarding the solution.*

## Problem Statement and Project Statement

The era of digital technology 4.0, along with the rapid development of the Internet, the trend of online business or online sales has brought economic efficiency to many business lines in Vietnam.

In recent years, "e-commerce" is no longer a strange concept in society or a new field in our country. One of them is an e-wallet. According to preliminary statistics, about 89% of consumers use e-wallets, 85% prefer to use credit and debit cards, mainly for online payment and contactless payment, online payment, banking, shopping, booking. We can take for example application platforms that use e-wallets such as Shoppe, Tiktok, Tiki...Create a shopping space for users with a lot of variety and save a lot of time.

But following the development of that online payment model, there are also many risks of information security in electronic transactions, such as payment fraud, phishing, ddos, … causes a lot of property damage to users. In order to enhance security and safety in online transactions, prevent threats, steal user data and assets, one of the ways to prevent that is to apply 2-factor authentication methods. 2FA), which is the SmartOTP method into an online application.

**Thesis objective**:

* Gain a comprehensive understanding of SmartOTP, a secure transaction authentication method for users in transactions with e-wallets. Create high security layers for user applications when making electronic payments and managing user transactions.
* Input: Personal information, OTP code.
* Output: Successful transaction.

## Context and Motivation?

### 2.1 Context.

OTP stands for **One-time password**, a unique password that is valid for a certain period of time and is a temporary cipher.

Smart OTP stands for **SMART One-time password**, Smart here means using smart methods to optimize the use of OTP codes during authentication and security. Smart OTP refers to an advanced implementation of a one-time password system for authentication purposes, a method of using advanced technologies and features to improve and facilitate the creation and use of OTPs. Here we integrate the OTP code right on the user application without having to go through any other 3rd party.

The **image below** describes the use of OTP authentication method, when the user receives the OTP code, enter it to authenticate the transaction.



*Figure 1: OTP verification example*

### 2.2 Motivation

To overcome limitations, improve security and user experience of traditional OTP methods, such as SMS OTP. Here are some key drivers for SmartOTP adoption:

* **Increased Security**: SmartOTP aims to provide a higher level of security compared to traditional OTP methods. By incorporating additional security factors such as dynamic images, audio, or facial recognition, SmartOTP reduces the risk of OTP theft, interception, or replay attacks. This motivation arises from the need to combat evolving cyber threats and protect sensitive user information.
* **User Convenience**: SmartOTP focuses on improving the user experience by offering a more convenient and user-friendly authentication process. Traditional OTP methods, like SMS OTP, require users to wait for and enter the code received via SMS. SmartOTP, with its multi-factor authentication options and advanced technologies, offers faster and more seamless authentication experiences, reducing user friction and enhancing convenience.
* Minimize SMS limitations: SMS OTP relies on SMS network availability and reliability, which can be prone to delays, network problems, or blocking. SmartOTP aims to mitigate these limitations by providing alternative authentication mechanisms that do not depend solely on sending SMS.
* **Enhance trust**: SmartOTP contributes to building trust and confidence among users by providing strong security measures and advanced authentication techniques.
* **Compliance with Regulatory Requirements**: Many industries and regions have specific regulatory requirements regarding user authentication and data protection. SmartOTP can be designed to meet these compliance standards, ensuring that organizations adhere to legal and regulatory frameworks.

## Compare Smart OTP and SMS OTP

|  |  |  |
| --- | --- | --- |
|  | Smart OTP | SMS OTP |
| Security | Has higher security than SMS OTP. SmartOTP uses a disposable OTP code in combination with other security elements face recognition. This reduces the risk of being attacked by stealing OTP messages | There is a risk of theft, as messages sent over cellular networks may be unsafe or delayed. |
| User Experience | Provides a better user experience than SMS OTP.Can use other authentication factors such as facial recognition, creating a faster and more convenient authentication experience for users. | Requires the user to wait and receive the message, then enter the OTP code into the app or website |
| Communicate | Does not depend on SMS messages to communicate with users, thus avoiding problems related to receiving messages | There may be cellular network-related issues, such as delayed messages, wrong delivery. |
| Multi-platform | Can be deployed on various platforms and devices but easier to integrate | Can be deployed on various platforms and devices |
| Availability and cost | May require significantly more deployment and integration, but may provide better availability and cost in the long run. | Available and common, and almost all modern mobile phones support receiving SMS messages. |

# Section 02: Background

***This section will explain the objective of building this project.***

## Expected Outcome

Create a modern, fast, online movie ticketing counter, save time and money. Ensure that users who book movie tickets are safe and secure, avoiding the risks of external agents by using SmartOTP authentication method.

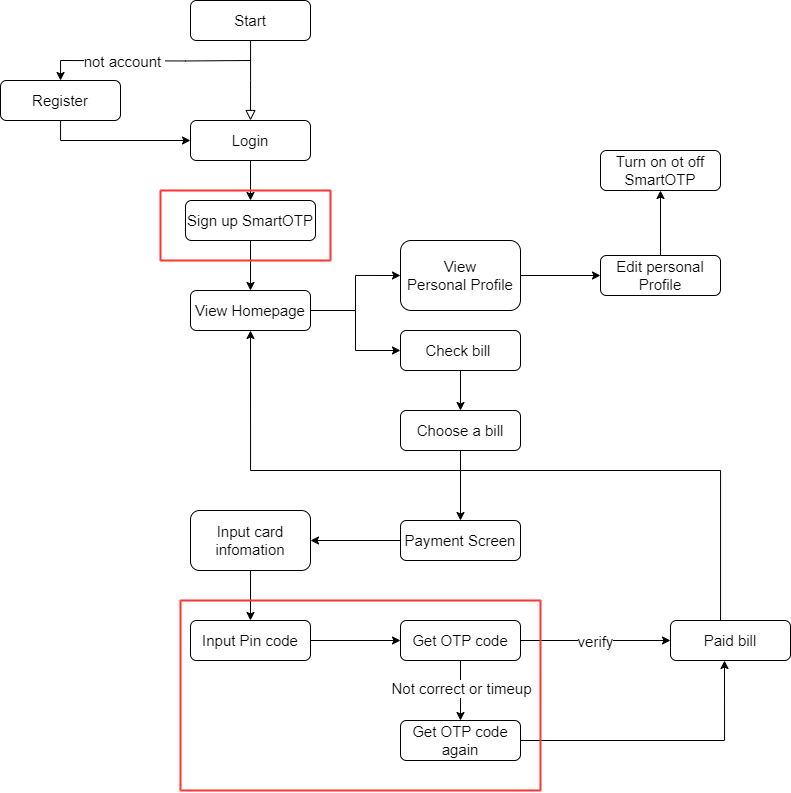
Breaking down into specific parts and steps, we divided all the tasks into phases:

* Phase 1: Designing and building a beautiful, user-friendly interface for mobile applications (running on both Android and IOS operating systems).
* Phase 2: Fulfill all the functions and requirements.
* Phase 3: Embedding a recommendation system.
* Phase 4: Fast authentication and verify OTP.
* Phase 5: Testing and transferring to End-user.

**In EVN, my contribution primarily focuses on Phase 1 and parts of Phase 2 (building User Interface some of features in this system).**

After 6 months of development, under our team process, the *application* External Attack Surface Management is in **half of Phase 4** of the pathway.

## Overall System



\

*Figure 2: Flowchart of application*

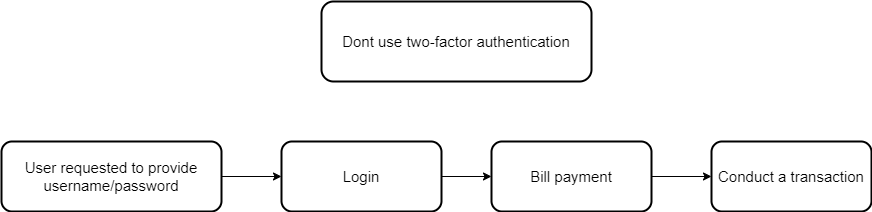
This is the flowchart diagram(or client view) of the application, it has basic features such as registration, login, payment. The red circled area is the important area of ​​the project, use registration and OTP authentication.

The overall features for the specific type of users is:

* For all users:
  + - Authentication & Registration. (Normal Priority)
      * + View & Update personal information. (Low Priority)
        + Upload personal data. (High Priority)
        + Create/Read/Update/Delete personal data. (High Priority)
        + Manage Workspace. (High Priority)
        + Data Interrogation via Queries insert. (High Priority)
* For administrators:
  + - Authentication & Registration. (Normal Priority)
      * + Users Management. (Normal Priority)
        + Update,Add,Delete films(Normal Priority)
        + Payment
* For OTP server:
  + - * + Authentication & Registration(High Priority)
        + Create Secret Key for users
        + Get Secret Key to decode into OTP code
        + Verify transaction

## 3. Two factor authentication service

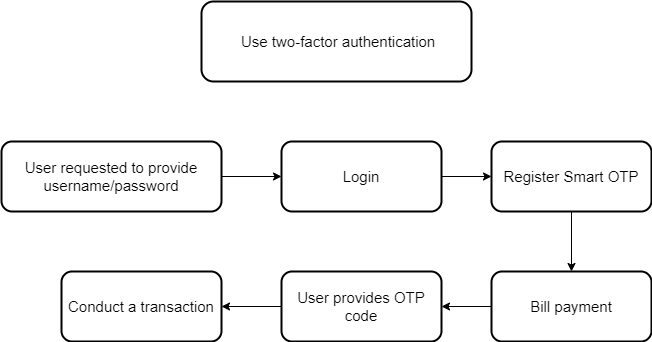
### 3.1. System does not use two-authentication factor.



*Figure 3: Verify without use 2-FA*

With the system not using two-factor authentication, users only need to register an account, login and pay the bill. There is only 1 layer of security when logging in, all of the following factors are not applied security layers.

### 3.2. System uses two-authentication factor.



*Figure 4: Verify using 2-FA*

With the use of Smart OTP service, the system has an extra layer of extremely safe security, users will have to provide an OTP code to provide to the system, after successful authentication, the system will allow it to be performed. successful transaction.

## 4. Smart OTP works

### 4.1. Register Smart OTP

A close-up of a blue sign

Description automatically generated

*Figure 5: Register Smart OTP*

This figure depicts first before using we must register the service first, after the user's registration request is sent to the service, the system will return a secret code to the user, each user has a unique code, which will be stored in both the client application and the service system. This code will generate OTP during verification.

### 4.2. Smart OTP verification

A close-up of a computer screen

Description automatically generated

*Figure 6: Smart OTP verification*

This figure depicts when the user starts to verify, the system starts decoding the user's secret key to generate an OTP code, after that, the OTP code is sent to the user, the user sends the code to the system for authentication, will there is 30s valid time from the time the code is generated, after that, the system will automatically generate a new code, forcing the user to request again.

### 4.3. OTP code generation

Currently, there are 2 algorithms born to generate OTP codes, which areHMAC-Based One-Time Password Algorithm (HOTP) and Time-based One-timePassword (TOTP), These are the 2 algorithms that googles application googleauthenticator is used to generate OTP codes.

**HMAC-Based One-Time Password Algorithm (HOTP)**

HOTP is an OTP password generation algorithm based on the SHA-1 hash

function.

About how to generate OTP password using algorithm:

HOTP: HOTP(K, C) = Truncate(HMAC\_SHA-1(K,C)) (1)

In formula (1):

K: Is the secret shared value between Client and Server

C: As the counter has been synchronized between Client and Server, C has a length of 8 bytes.

Truncate (): Is a string splitter function that extracts the results from the Hash function to get the OTP password.

The method to split the string is as follows:

HMAC\_SHA-1(K, C) =SHA-1(K⊕C1… ∥ SHA-1(K⊕ C2… ∥ C)) (2)

In formula (2):

C1 = 0x36 (36 in hexadecimal).

C2 = 0x5c (5C in hexadecimal).

HMAC\_SHA-1(K, C) is a computation function based on HMAC algorithm combined with SHA-1 hash of value K and counter C.

The password value is calculated according to the following formula:

= HOTP(K, C) mod (3)

In formula 3: d is the number of digits of the OTP, usually a generated OTP.

password has a length of 6 to 8 digits (e.g. 123456, ...).

**Role**: HOTP generates an OTP based on a combination of a secret key and a unique number. When the user requests the OTP code, the system will use the current unique number to calculate the OTP code, and the unique number will increase after each use to ensure the uniqueness of the code.

**Processing process:**

* User asks for OTP code.
* The system uses the HMAC function to calculate the OTP from the current secret key and unique number.
* The OTP is returned to the user.

**Time-Based One-Time Password Algorithm (TOTP)**

With the TOTP algorithm generating OTP passwords based on time, the T

value of time is calculated as follows:

In formula (4):

is the current time value calculated in Unix time (calculated from the time of the Unix Epoch is 01/01/1970 UTC (universal standard time).

T0: Is the initial time value (usually choose T0 = 0).

X: As the time step, this is the parameter that determines the valid time of the OTP password.

T is the calculation result (taken the integer part) from the above calculation formula.

With the TOTP algorithm, the standard valid time of the OTP password is 30 seconds (X=30), the valid time of each use of the OTP password is 30 seconds, chosen in accordance with the security requirements and usability.

The TOTP algorithm is based on the HOTP algorithm that replaces the count value (C) with the time value (T):

TOTP = HOTP(K, T) (5)

For the TOTP algorithm, the length of the OTP password is calculated as follows:

= TOTP(K,T) mod (6)

In formula (6): d is the number of digits of the OTP password, usually a generated OTP password has a length of 6 to 8 digits (e.g. 123456, ...).

**Role**: TOTP generates an OTP based on a combination of the secret key and the current time. When the user requests the OTP code, the system will use the current time to calculate the OTP code, and the OTP code will change over time cycle (e.g. every 30 seconds).

**Processing process:**

* User asks for OTP code.
* The system uses the current time and the HMAC function to calculate the OTP from the secret key and the time.
* The OTP is returned to the user.

**Compare HOTP with TOTP**

|  |  |  |
| --- | --- | --- |
|  | HOTP | TOTP |
| Uniqueness | Based on incrementing unique numbers. | Base on time |
| Time | Independent of time | Will change over time, help improve security |
| Security | Can be more secure in an environment where time can be hacked | Vulnerable to timing attacks (computer clocks are interfered with) |
| Storage time | Can be deployed on various platforms and devices but easier to integrate | Need to update sync with the computer or device used |

**Why not use a random number?**

Using HOTP and TOTP instead of a random sequence of numbers is simply for the sake of security and convenience when authenticating users in online systems. Random numbers can be easily predicted or hacked, while HOTP and TOTP use encryption and timing methods to generate more secure OTPs. Here are some specific reasons:

* **More convenient and portable**: HOTP and TOTP can be easily integrated into mobile applications or hardware devices such as smart cards. Users can easily receive and enter OTP codes when needed, instead of having to carry or memorize a random sequence of numbers.
* **Reusability and Convenience**: HOTP and TOTP allow users to use the same secret key to generate multiple OTPs. This makes daily or weekly OTP changes unnecessary, reducing code loss.
* **Time security**: TOTP uses time as an additional factor to generate OTP codes. This means that the attacker must know both the secret key and the exact time to generate a valid OTP.
* **Combine Convenience with Security**: HOTP and TOTP combine cryptographic randomness and timing to generate OTP codes, creating a powerful method for user authentication.

**We use TOTP instead of HOTP!**

**Timeliness and convenience**: TOTP uses time as an additional factor to generate OTP codes, making the authentication process easier for users. The OTP code changes every fixed interval (e.g. every 30 seconds), eliminating the need for users to enter new codes too often and also ensuring greater security.

**More Modernity and Popularity**: TOTP is a common and standard authentication method, widely supported by many mobile applications and services. This makes TOTP integration and deployment easier and more flexible.

**Attack response**: TOTP can respond better to a wide range of attacks. Since OTP codes change over time, it becomes more difficult for an attacker to predict and reuse the code.

**Auto-Sync Capability**: Since TOTP uses common time, it can be easily synchronized across multiple devices. Users can install the same TOTP application on multiple devices and the OTP code will automatically change on these devices simultaneously.

### 6.4 Security capabilities of the system

* **Secure information with RSA algorithm**

**A screenshot of a computer

Description automatically generated**

*Figure 7: RSA algorithm*

The RSA algorithm (Rivest-Shamir-Adleman) is the basis of a cryptosystem -- a suite of cryptographic algorithms that are used for specific security services or purposes -- which enables public key encryption and is widely used to secure sensitive data, particularly when it is being sent over an insecure network such as the internet.

Public key cryptography, also known as asymmetric cryptography, uses two different but mathematically linked keys -- one public and one private. The public key can be shared with everyone, whereas the private key must be kept secret. In RSA cryptography, both the public and the private keys can encrypt a message. The opposite key from the one used to encrypt a message is used to decrypt it.

**When the user registers Smart OTP**, the system will use 1 public key of the user to encrypt and return to the user an undecrypted secret key, each user will have 1 public key and 1 secret key, **when the user OTP authentication**, the system will decrypt that secret key with a private key to generate a decrypted code, then use the **TOTP algorithm** to calculate this code into a sequence of OTP codes, then sent to the person for authentication, the user uses this OTP code and a JWT token code to verify. All these actions I have tested successfully on the web and are still in the process of integrating into the application.

* **Authentication using JWT (Json Web Token) method.**

**A key with a key and a sign

Description automatically generated with medium confidence**

*Figure 8: Json Web Token*

It is a method of exchanging information between the user side and the service system, it acts as a key, helping to ensure the integrity of the data, ensuring it is not tampered .

A screenshot of a computer

Description automatically generated

*Figure 9: OTP verification*

After we have obtained the OTP code when the user enters and JWT, the system will proceed to authenticate, if the factors are correct, the transaction is successful, if not, proceed again.

# Section 03: Analysis and Design

*This section will explain and demonstrate all the processes and functions that are expected to be executed online with our application.*

## System Architecture

A screenshot of a computer

Description automatically generated

The chart above depicts the system architecture of the Smart OTP application, a project with large storage capacity and high security requirements. The hierarchical system and database storage need to be carefully planned. The system architecture comprises the following main components:

* **Customer, user:** users use the application.
* **Client system:** Customer application, where customers operate and use.
* **Server system:** Application for administrators, functional changes, and application updates
* **Payment service:** receives information from users, verifies it with the bank to check the customer's assets.
* **Smart OTP service:** a third-party service used for payment authentication**.**

The system architecture is divided into two main parts:

* **Register to use Smart OTP:**

A screenshot of a computer

Description automatically generated

*Figure 10: System architecture of register OTP*

This drawing describes the system architecture of Smart OTP, how it works in two modules: registration and verification. When a user registers to use the service, the system will send a registration request to the Smart OTP service server, from the user code, the system will use the RSA algorithm to generate a secret code for the user. This secret will be saved on the service side, and also sent to the user application, the two parties are able to communicate and exchange with each other through Rest APIs - an application used to transform the structure of data that can be stored. methods to help connect with different libraries and applications.

* **Smart OTP verification**

**A screenshot of a computer

Description automatically generated**

*Figure 11: System architecture of verify OTP.*

When the user authenticates the transaction upon payment, the client application will get the secret code provided by the Smart OTP service. When registering to decrypt, the code after being solved will be used by the TOTP algorithm. To generate OTP code, the Smart OTP service also does the same, decoding the secret code and generating OTP code. After that, the user will send this OTP code to the service to check whether the value is correct or not. The code's validity period is 30 seconds, after 30 seconds, the service system will automatically generate the code. new, which causes the user to re-enter the transaction. After successful validation, the result is correct, the transaction is successful, and the invoice is returned to the user.

## Use Cases

Use case diagrams are used to describe the functionality of a system using actors and features. Use cases specify the expected behavior. In this scenario, the “Smart EVN” is a mobile application.:

A screen shot of a diagram

Description automatically generated

*Figure 12: Use Case Diagram*

## User Characteristics

There are 4 stakeholders that we consider interacting with the system the most: Admin, User, Smart OTP service and Payment System. Each stakeholder has a different background, so each of them has its requirements for the system:

* **Admin**: People who oversee making modifications deeply in the system should have an easy-to-use User Interface, and not-too-much technical functionalities as the objective of this application is simple, but effective.
* **User**: Permissions that can be exercised to the user interface system.
* **OTP service**: OTP authentication system, encrypt secret key to generate OTP code, send to user to authenticate transaction, each OTP code is valid for 30 seconds.
* **Payment System**: This system of admin is used to pay for transactions after each ticket booking, after customers choose tickets, and make payment by credit card.

## Use Cases and Scenario Description

In this part, only use cases with a certain complexity that generate more than one system action or need compulsory data will be mentioned. They are:

* Module: Register and Login (For all users)
* Module: Register Smart OTP. (For all users)
* Module: Payment. (For all use)
* Module: Verify OTP.

### 4.1. Module: Register and Login

* Brief Description: This use case depicts how an actor log in or sign up to their account
* Basic Flow:

|  |  |  |
| --- | --- | --- |
| **User Action** | **System Action** | **Data needed** |
| 1. Actor accesses the application | 2. System returns Welcome screen with Login/Sign Up button |  |
| 3. Actor chooses to Login or Sign Up | 4. System returns “Login/Sign Up” button based on actor’s choice | Based on the actor's choice. |

* Alternative Flow: At step 3
  + Scenario 1: The actor chooses to Log in.

| **User Action** | **System Action** | **Data needed** |
| --- | --- | --- |
| 3. Actor choose to log in by hitting “Login” button | 4.  System returns Login screen |  |
| 5. Actors fulfill all the needed information & hit “Login” button | 6. System returns home screen |  |

* Scenario 2: The actor chooses to Sign Up

|  |  |  |
| --- | --- | --- |
| **User Action** | **System Action** | **Data needed** |
| 3.  Actor choose to sign up by hitting “Sign Up” button | 4.  System returns Login screen |  |
| 5.  Actor fulfill all the needed information & hit “Sign Up” button | 6.  System checks for the information. | LOGIN/SIGNUP |

* Scenario 3: Step 6 of Sign-Up process.

|  |  |  |
| --- | --- | --- |
| **User Action** | **System Action** | **Data needed** |
|  | 6. If the actor’s information is coincident with existed information, system returns errors and make actor to re-enter information |  |
| 7.  Actor re-enter the coincident information & hit “Sign Up” button | 8. System rechecks the information if there is any coincident information. If not, system sends the validate code to actor’s email | LOGIN/SIGNUP |
| 9.  Actor enter the validate code & hit “Confirm” button | 10.  System returns Home screen |  |

* Scenario 4: Step 6 of Sign-Up process.

|  |  |  |
| --- | --- | --- |
| **User Action** | **System Action** | **Data needed** |
|  | 6. If the actor’s information is not coincident with existed information, system returns validate code to actor’s email | LOGIN/SIGNUP |
| 7.  Actor enter the validate code & hit “Confirm” button | 8.  System returns Home screen |  |

* Special requirements: For actor’s input information
  + Email must be valid address.
  + Username must include both characters & numbers.
  + Password should include both uppercase, lowercase, and special characters.

### 4.2. Module: Register Smart OTP

* Brief description: This use case depicts users register for Smart OTP authentication method, ensuring safety in transactions.
* Basic Flow:

|  |  |  |
| --- | --- | --- |
| **User Action** | **System Action** | **Data needed** |
| Users press the "register now" button on the homepage of the application | System show Welcome for smart otp screen |  |
| User generates a passcode to unlock the transaction authentication method | System shows input Pin code and then give Secret key for user and save it | Information of users (email, phone number) |

* Alternative Flow:
  + Since this is a required method for safety, there will be no alternative flow.
* Special Requirements:
  + Pin code is a 6-digit string of digits.

### 4.3. Module: Payment

* Brief description: This use case depicts the user will enter credit card information to pay the bill.
* Basic Flow:

|  |  |  |
| --- | --- | --- |
| **User Action** | **System Action** | **Data needed** |
| User input card credit | Check card information, give full information to pay. | dataset |

* Alternative Flow:

|  |  |  |
| --- | --- | --- |
| **User Action** | **System Action** | **Data needed** |
| User input wrong Card number | Re-enter. | dataset |

* Special Requirements: Enter the correct card credit information.

### 4.4. Module: Verify OTP

* Brief description: This use case depicts the server has the right to add, edit, delete, update movies on the application.
* Basic Flow:

|  |  |  |
| --- | --- | --- |
| **User Action** | **System Action** | **Data needed** |
| User input Pin code | sent secret key to encode, give OTP code for user application. | dataset |
| User input OTP code | check the OTP code if it is the same as on the server, if it is the same, the authentication message is successful. | dataset |

* Alternative Flow:

|  |  |  |
| --- | --- | --- |
| **User Action** | **System Action** | **Data needed** |
| User input wrong Pin code | reported an error, asking to re-enter. | dataset |
| User entered OTP code timed out | request to rebook the ticket from the beginning. | dataset |

* Special Requirements: Enter the correct Pin code, OTP code as given before

### 4.5. Expected Features

The list below shows this mobile application's core expected features.

**For all users:**

* **Bill:** check the most accurate information about electricity bills that users have consumed within 1 month**.**
* **Service registration:** users can register for the service, issue the code accurately and quickly.
* **Payment**: fully check the information entered by the user, return the correct credit card information, the correct amount in the card.
* **Validate OTP**: provide OTP code to users, check codes when users submit and return accurate, fast, and secure results.

### 4.6. Executed Features

The list below shows this mobile application's expected features.

**For all users:**

* **Authentication and registration**:
  + Login.
  + Two-factor authentication.
  + Integration.
* **Register Smart OTP:**
  + Get secret key.
  + Create Pin code.
* **OTP verification**:
  + Get JWT.
  + Create OTP code.

## Sequence Diagram

This section will include the sequence diagram of prominent features within the system.

### 5.1. System Authentication.

A screenshot of a computer

Description automatically generated

*Figure 13: Register sequence diagram*

A screenshot of a computer screen

Description automatically generated

*Figure 14: Login sequence diagram*

This use case describes how a user logs in to the system. When a user chooses to Login, the system will return a Login Form that requires the user to enter their information, which is the registered email address and password. While finishing entering the information and hitting enter, the system will validate the input information data. If there are no problems with the input information, the system will let the user pass through. If there are problems, the system will return to the Login form and require the user to re-enter the information correctly.

### 5.2. Check bill.

A screenshot of a computer screen

Description automatically generated

*Figure 15: Check bill sequence diagram*

The figure above describes the system sequence of viewing bills to pay. When the user accesses the homepage, selects the bill to pay, the system will return all information about the electricity bill that needs to be paid math.

### 5.3. Sign up Smart OTP

A screenshot of a computer

Description automatically generated

*Figure 16: Sign up Smart OTP sequence diagram*

The figure above describes the system sequence of registering to use Smart OTP service, when the user provides information, the system will receive and authenticate, save, and return to the user a secret code to perform. payment process.

### 5.4. OTP Verification

A screenshot of a computer

Description automatically generated

*Figure 17: OTP verification sequence diagram*

The figure above depicts the system sequence of Smart OTP payment authentication. When the user enters the Pin code to unlock the process, the client application and the service system will generate an OTP code, the customer sends the OTP code. for the system to authenticate, along with the JWT code after it is generated, if the OTP code value is correct, the authentication process is successful, if it is wrong, re-enter the code every 30 seconds.

### 5.5. Payment

A screenshot of a computer screen

Description automatically generated

*Figure 18: Payment sequence diagram*

The figure above depicts the system sequence of payment method selection, here the default is credit card. When the user enters card information, the system will link with the bank to check all the user's card information and return it to the user for the user to enter the amount. If incorrect information is entered, proceed. retype.

## 6. Functional requirements

**1. Cloud-based**

* Provide payment services via the Internet.

**2. Payment option**

* Allow users to use credit cards and bank accounts to pay.

**3. User-friendly interface**

* The application has a beautiful interface, convenient, easy to use, not confusing for beginners.

**4. Customer support**

* Quickly solve customer problems during the booking process

**5. Create and sent OTP:**

* Generate a very secure one-time OTP password based on the TOTP algorithm.
* Send it to the user for authentication after encrypting the secret key.

**6. OTP verification**

* The system will verify the OTP with the server's OTP after user request authentication to ensure safety and authorization.

**7. User Management:**

* The system should provide functionality to manage user accounts, including user registration, password reset, and account deactivation.
* Integration
* The system should be able to integrate with other systems or applications, such as user databases, authentication servers, or third-party services.

## 7. Non-functional requirements

* **Security**: Smart OTP system uses a strong encryption algorithm to protect OTP transmission and storage. The Smart OTP server takes measures to prevent unauthorized access or OTP spoofing. The Smart OTP system has mechanisms in place to detect and prevent fraud or abuse related to OTP.
* **Performance**: The Smart OTP verification process has minimal latency to ensure a seamless user experience.
* **Scalability:** The Smart OTP system is designed to accommodate the growing user base and growing OTP verification. Smart OTP server architecture will allow easy scaling.
* **Reliability:** The Smart OTP system is highly available to ensure that OTP verification is always accessible to users. Resilient to unexpected failures.
* **Usability**: Smart OTP integration in the booking application has a user-friendly interface and clear OTP input instructions, timely and accurate error message is required when the user cannot verify the OTP.
* **Compatibility**: The Smart OTP system is compatible with many mobile devices and operating systems to serve a wide range of users.

## 8. Problem arise

**Problem:** When the user uses OTP to authenticate the transaction for order 1 (OTP code is valid for 30 seconds), but after receiving the code (without pressing the authentication button), the user wants to cancel that order and return. home page and select another order (the OTP code's validity period of 30 seconds has not yet expired), so when the user pays for that order, how will the OTP code be processed?

**Answer:** This case involves managing OTP codes when users make changes or cancel transactions while the OTP code is still valid. Here's how the system might handle the situation you described:

1. When receiving OTP code but not authenticating:

* In case you receive the OTP code for order 1 but have not pressed the authentication button, the OTP code is still valid for 30 seconds.
* If you decide to cancel order 1 and return to the home page to select another order, the system can record the cancellation of order 1 and disable the OTP code related to this order.

1. When making a new transaction:

* When you make a new transaction for another order and want to use the OTP code, the system can generate a new OTP code for this new transaction.
* This new OTP will have its own validity and will not be affected by the old OTP.

1. Processing OTP code when paying:

* When you make payment for a new order and use the new OTP code, the system will authenticate this new OTP code to ensure transaction security.
* The old OTP from the previous order has been canceled and will no longer be usable to authenticate any transactions.

In short, the system needs to manage OTP codes flexibly to ensure safety and reliability for users. In case you cancel your order and make a new transaction within the time period when the OTP is still valid, the system should generate a new OTP for the new transaction to avoid confusion and ensure that the old OTP cannot be used. used after the transaction has been cancelled.

## 9. Database diagram

**A screen shot of a computer code

Description automatically generated**

*Figure 19: Database diagram*

The above diagram describes the entire data of the project, there will be 3 main items that are users, invoices, and OTP system. Here is the detailed description.

### Object details

This is the user's data table, there will be user code, name, account, password, address:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User** | | | | |
|  | Attribute | Datatype | Constraint | Description |
| 1 | user\_id | number | PK | 0343543,0545484 |
| 2 | user\_name | string |  | usth1234 |
| 3 | user\_email | string |  | usth123@gmail.com |
| 4 | user\_password | string |  | 123321 |
| 5 | user\_address | string |  | Yen Hoa,Cau Giay,Ha Noi |

*Table: user table*

This is the OTP data table, it will use the user's code, there will be have appCode, appKey, OTP code and JWT code:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OTP** | | | | |
|  | Attribute | Datatype | Constraint | Description |
| 1 | user\_id | number | FK | 0343543 |
| 2 | appCode | string |  | HFUR43 |
| 3 | appKey | number |  | 43737 |
| 4 | OTPcode | number |  | 123321 |
| 5 | JWTcode | string |  | Fhdfhdjfhdfhsdkjf40958343rhd |

*Table: OTP table*

This is the OTP data table, it will use the user's code, there will be have bill\_id, user code, user name,user account, user address,number\_electricity and total\_fee:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bill** | | | | |
|  | Attribute | Datatype | Constraint | Description |
| 1 | bill\_id | number | PK | 23275847583 |
| 1 | user\_id | number | FK | 0343543 |
| 2 | user\_name | string |  | usth1234 |
| 3 | user\_email | string |  | usth123@gmail.com |
| 4 | user\_password | string |  | 123321 |
| 5 | user\_address | string |  | Yen Hoa,Cau Giay,Ha Noi |
| 6 | date | date |  | 21/12/2023 |
| 7 | number\_electicity | number |  | 232 |
| 8 | total\_fee | number |  | 145.000 |

*Table: bill table*

## 10. Tools

### 10.1. Materials Distribution

* **Mobile Platforms**: React Native
  + User Interface
  + Functionalities
* **Serving**: MariaDB
  + Request & Return
  + Database Storing Server in: MariaDB.
    - Request & Return
    - Database Storing

### 10.2. Tools

**1. Android Studio**

A green square with a black and white compass

Description automatically generated

*Figure 20: Android Studio logo*

I use this software to emulate android, to run virtual machines on windows operating systems, to help integrate applications on computers.

**2.MariaDB**

**A white seal with blue text

Description automatically generated**

*Figure 21: MariaDB logo*

I use this software to store data, user data, OTP code, and secret key.

**3. IntelliJ**

A logo with colorful triangles

Description automatically generated with medium confidence

*Figure 22:Intellij*

This is a famous software for programmers, me and my mentor use it to build backend structure for the project. I use it to code the Java language.

**4. Visual Studio Code**

A blue ribbon with a cross

Description automatically generated

*Figure 23:Visual studio code*

This is software to code for programmers, I use it to build user interfaces, use javascript language, and the support library is React native.

**5. React Native**

A blue and black symbol

Description automatically generated

*Figure 24:React native*

With React Native, I create cross-platform mobile apps that can run on both IOS and Android platforms, sharing a significant portion of code between the two.

**6. Postman**

A white figure in a helmet

Description automatically generated

*Figure 25:Postman*

I design, test, and document APIs (Application Programming Interfaces). It provides a user-friendly interface and a comprehensive set of features for simplifying the process of working with API

# Section 04: Results

## Results

After nearly 6 months of planning and development, we have also basically built the user interface of the electricity bill payment application of EVN. The project is almost complete in terms of documentation, design, user interface and necessary basic services. We have conducted rigorous and thorough testing and inspection.

During the testing phase, we diligently tested the functionality and performance of the system, ensuring that it met the highest standards. We have performed extensive test scenarios to identify and resolve any potential vulnerabilities or issues.

Now that the project has reached this stage, we are proud to have achieved an important milestone, providing users with more safety and security in bill payments, in the future, we will complete them as quickly as possible.

Project State:

* Project run: 75%.
* Design: done
* User interface: 90%.
* SRS Document: done.
* Testing(private)

## Result of my contribution

In this project, my main role was designing and building the user interface for the application. I also participated in developing the project's backend, creating APIs, and connecting the customer application with the services. While my tasks weren't very challenging, during my internship and work at EVN Corporation, I learned a lot about work processes, teamwork, and gained knowledge in various programming languages. I have gained more experience in my field of expertise.

Contribution in Front-end Developer role:

* Create components.
* User Interfaces (done)
* Testing(continuing)

## 3. Product Preview

Please have a look at the preview of the product in the **Annex Section** of this document. (***Page 35***)

# Section 05: Conclusion and Future Work

## Conclusion

After 6 months of internship, in this project, I have gained valuable knowledge about cyber threats, e-wallet transaction attacks, user scams, understanding of secure code generation algorithm, multiple programming languages, journey from beginner to expert, and development, execution and deployment of a mobile app. During the project, I received tremendous support and encouragement from my team members as well as my team leader, Doan Van Long. Starting with just an idea, we worked hard for months to turn it into a highly functional app. I am so grateful for the opportunity to be a part of this project, the teamwork, dedication, and attention to detail shown throughout the project were key factors to its success of project. At the conclusion of this project, we look forward to future efforts in the field of cyber security, transaction security, using the knowledge and skills gained from this experience to continue to make valuable contributions. value in protecting digital assets and creating a user-friendly environment, bringing good experiences to users.

I am incredibly grateful for the opportunities this project has provided me in the field of Information Technology and Cyber Security Department. At the beginning of my internship, my experience in developing React Native applications with real databases was minimal. However, through collaboration with my teammates, I have acquired the skills necessary to independently develop an application. I am indebted to my supervisors, Dr. Doan Nhat Quang and Dr. Doan Van Long, as well as all the members of my project team, for their guidance and support.

Although my internship at the EVN has come to an end, I believe there is still untapped potential for this application. I look forward to exploring further possibilities and expanding its capabilities in the future.

## Future Work

The up-coming features of the application:

* Complete all features of OTP system services such as providing OTP codes.
* Build your own payment feature through scanning code, can pay for any bill.
* Integrate Face ID into OTP unlocking, instead of Pin code.
* Build the web version of the project.

# References

|  |  |
| --- | --- |
| [1] | [Online]. Available: https://store.steampowered.com/agecheck/app/1091500/?l=vietnamese. |

# Annex

Below are the interface figures of the system:

A screen shot of a phone

Description automatically generated

*Figure 26: Loading & Login User Interface*

Figure 26 shows the interfaces of Application Loading and User Login screen.

A screen shot of a phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

*Figure 27: Register Smart OTP*

Figure 27 shows the Dashboard Interface and Menu Interface. Here, users can have a quick view of their favorite information. Also, they can have an overview about vulnerabilities of their assets.

A screen shot of a phone

Description automatically generated

*Figure 28: Home Screen Application*

Figure 28 shows the Uptime Overview Interface. Here, user can have an overview of list assets in uptime service. Also, user can search, filter, on/ off scanning asset.

A phone with numbers and a blue button

Description automatically generated

*Figure 29: OTP screen*

Figure 29 shows the Uptime Detail Interface. Here, user can have an overview of one asset in uptime service. Also, user can filter the status monitoring by chart heatmap, response time by line chart, view SSL Certificate.

***Final Fage***