VIETNAM-KOREA UNIVERSITY OF INFORMATION AND  
COMMUNICATION TECHNOLOGY

**Faculty of Computer Science**



GRADUATION THESIS

**ABC**

Student: **LE THANH AN**

**LE VAN HIEU**

Class: **18IT5**

Supervisor: PH.D NGUYEN HUU NHAT MINH

***Da Nang, December – 2023***

VIETNAM-KOREA UNIVERSITY OF INFORMATION AND

COMMUNICATION TECHNOLOGY

**Faculty of Computer Science**

GRADUATION THESIS

**ABC**

Student: **NGUYEN VAN A**

Class: **18IT5**

Supervisor: NGUYEN VAN B

***Da Nang, December - 2022***

SUPERVIOR’S COMMENT

ACKNOWLEDGEMENTS

We would like to profoundly acknowledge the people who have helped me during my studies:

First of all, we would like to express my heartfelt gratitude to my family for their unwavering support and encouragement throughout my academic journey

We would like to express my sincere gratitude to my thesis advisor, Ph.D. Nguyen Huu Nhat Minh, for his invaluable guidance and support throughout the completion of my thesis. His expertise and encouragement have been instrumental in the successful development of my research on "Applying Blockchain in Building a Charitable Crowdfunding Website."

We would also like to extend my thanks to Vietnam-Korea University of Information and Communication Technology for providing me with the opportunity to pursue my academic endeavors.

Finally, We express gratitude to myself for the dedication, hard work, and perseverance invested in completing this thesis. The challenges were significant, but the personal growth achieved in the process was invaluable.

STATEMENT OF AUTHORSHIP

*Danang, December 5th, 2022*

NGUYEN VAN A

TABLE OF CONTENTS

[SUPERVIOR’S COMMENT iii](#_Toc153519877)

[ACKNOWLEDGEMENTS iv](#_Toc153519878)

[STATEMENT OF AUTHORSHIP v](#_Toc153519879)

[TABLE OF CONTENTS i](#_Toc153519880)

[ABBREVIATIONS iv](#_Toc153519881)

[LIST OF FIGURES v](#_Toc153519882)

[LIST OF TABLES vi](#_Toc153519883)

[INTRODUCTION 1](#_Toc153519884)

[1. Problem statement 1](#_Toc153519885)

[2. Aims and Objectives 1](#_Toc153519886)

[3. Structure of the thesis 1](#_Toc153519887)

[CHAPTER 1. THEORETICAL FOUNDATION 2](#_Toc153519888)

[1.1 Blockchain and smart contract overview 2](#_Toc153519889)

[1.1.1 Blockchain introduction 2](#_Toc153519890)

[1.1.2 Blockchain application 3](#_Toc153519891)

[1.1.3 Advantages and Disadvantages of Using Blockchain 3](#_Toc153519892)

[1.1.4 Smart contract introduction 5](#_Toc153519893)

[1.1.5 How smart contract works 5](#_Toc153519894)

[1.1.6 Benefits of smart contracts 5](#_Toc153519895)

[1.2 DAPP (Decentralized Application) overview 6](#_Toc153519896)

[1.2.1 DAPP (Decentralized Application) introduction 6](#_Toc153519897)

[1.2.2 Properties of dApps 6](#_Toc153519898)

[1.2.3 Centralized App and Decentralized App Comparison 6](#_Toc153519899)

[1.3 Technologies is used in front-end side 8](#_Toc153519900)

[1.3.1 Typescript definition 8](#_Toc153519901)

[1.3.2 Components of TypeScript 8](#_Toc153519902)

[1.3.3 Why use TypeScript 8](#_Toc153519903)

[1.3.4 React library introduction 9](#_Toc153519904)

[1.3.5 How does React works? 9](#_Toc153519905)

[1.4 Technologies is used in back-end side 9](#_Toc153519906)

[1.4.1 Nodejs introduction 9](#_Toc153519907)

[1.4.2 Express.js 10](#_Toc153519908)

[1.5 NoSQL and MongoDB overview 10](#_Toc153519909)

[1.5.1 NoSQL definition 10](#_Toc153519910)

[1.5.2 Difference between RDBMS and NoSQL databases 10](#_Toc153519911)

[1.5.3 MongoDB Definition 10](#_Toc153519912)

[1.5.4 Features of MongoDB 10](#_Toc153519913)

[1.5.5 Nodejs and MongoDB combination 11](#_Toc153519914)

[CHAPTER 2. SYSTEM ANALYSIS AND DESIGN 12](#_Toc153519915)

[2.1 System requirements 12](#_Toc153519916)

[2.1.1 Overview about project system 12](#_Toc153519917)

[2.1.2 Functional requirements 12](#_Toc153519918)

[2.1.3 Non-functional requirements 12](#_Toc153519919)

[2.2 UML modeling diagrams 12](#_Toc153519920)

[2.2.1 Use-case diagram 12](#_Toc153519921)

[CHAPTER 3. WEBSITE CONSTRUCTION AND RESULT 14](#_Toc153519922)

[3.1 Insert title here 14](#_Toc153519923)

[3.2 Insert title here 14](#_Toc153519924)

[3.3 Summary 14](#_Toc153519925)

[CONCLUSIONS AND SUGGESTIONS 15](#_Toc153519926)

[1. Conclusions 15](#_Toc153519927)

[2. Suggestions 15](#_Toc153519928)

[REFERENCES 16](#_Toc153519929)

[PUBLICATIONS 17](#_Toc153519930)

[APPENDIX 18](#_Toc153519931)

ABBREVIATIONS

|  |  |
| --- | --- |
| **ABBREVIATIONS** | **MEANING** |
| 5E | Engagement, Exploration, Explanation, Elaboration, Evaluation |
| STEM | Science Technology Engineering Mathematics |

LIST OF FIGURES

[Figure 1.1 - VKU icon 2](#_Toc114744561)

LIST OF TABLES

[Table 1.1 - Related component-based approaches to the defined challenges 2](#_Toc114744541)

INTRODUCTION

##### Problem statement

##### Aims and Objectives

##### Structure of the thesis

After the *Introduction*, the thesis is structured in three chapters:

*Chapter 1,* *.................* In this chapter, the thesis presents …...........

*Chapter 2, ................*. This chapter includes ...........

*Chapter 3, ................* This chapter proposes…......

Finally, there are *Conclusions*, *Suggestions,* *References* and *Appendices* related to the topic.

# THEORETICAL FOUNDATION

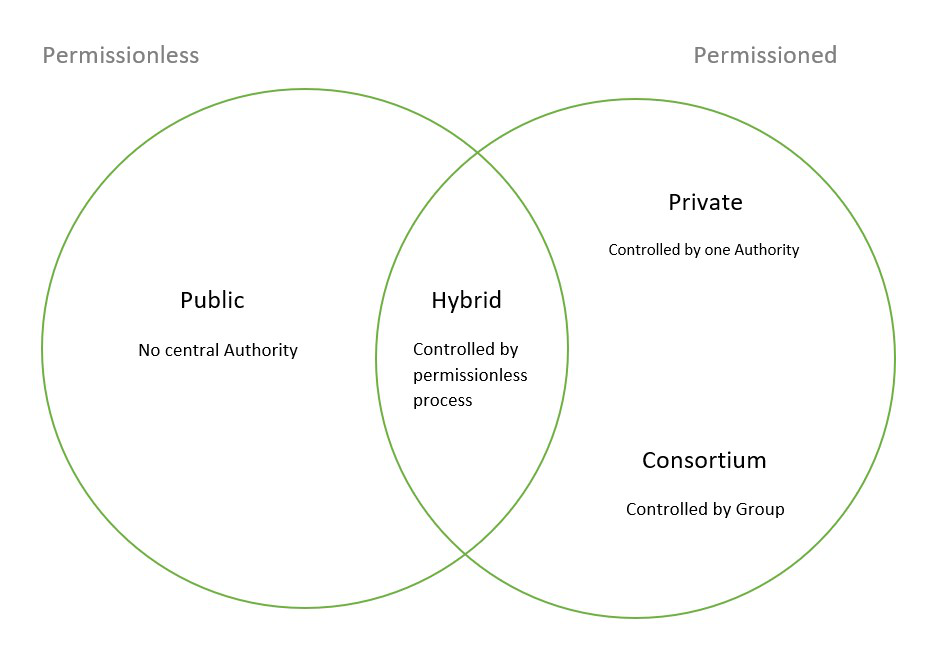
## Blockchain and smart contract overview

### Blockchain introduction

Blockchain is also known as the chain technology - similar to a digital public ledger, where every transaction is recorded and closely monitored on a decentralized network, without the intervention of third parties, fully automated. The information blocks in the Blockchain are called "blocks" and allow for expansion over time. Each block contains information about the initiation time and is tightly linked to the previous block. Data loaded into a block, once accepted by the network, cannot be altered. Blockchain is designed to resist fraud and data tampering.

Classification of Blockchain:

* Public Blockchain: Public blockchains are open and permissionless networks where anyone can join, participate, and validate transactions. They operate on a decentralized model, allowing for transparency and security. Examples include Bitcoin and Ethereum.
* Private Blockchain: Private blockchains are restricted to a specific group of participants, and access permissions are controlled. These are often used within organizations or consortia, providing a more centralized control structure. Privacy and efficiency are emphasized in private blockchains. Examples include Hyperledger Fabric and R3 Corda.
* Hybrid Blockchain: Hybrid blockchains combine elements of both public and private blockchains. They allow for a certain degree of openness and decentralization while also providing controlled access to a select group of participants. This flexibility is beneficial for various use cases, offering a balance between transparency and privacy.



Popular blockchain platforms include:

* Bitcoin: The first and most widely used blockchain platform for cryptocurrency transactions. Bitcoin is utilized as a digital currency for online transactions and serves as a foundation for the development of other financial applications.
* Ethereum: The most popular blockchain platform for developing non-financial applications, with smart contracts being the primary technology for creating these applications.
* Solana: A blockchain platform known for its fast transaction speed and low fees. Designed to support non-financial applications and cryptocurrencies, Solana stands out for its robust transaction processing capabilities and high performance.

### Blockchain application

There are various fields where Blockchain can be applied, each bringing unique approaches. Here are some examples:

* Manufacturing: Blockchain can transparently track and monitor the production and distribution processes of products. It provides information about inventory quantity and quality, transaction details, and the ability to trace the origin of products.
* Healthcare: In the healthcare sector, Blockchain can help create a network between applications managing medical information and ensure the quality of healthcare. It has the potential to improve data management and discover critical information.
* Education: Applying Blockchain in education can minimize issues related to fraud in the learning process, scholarship applications, job searches, and even verification of educational qualifications and work experience.
* Banking: In the financial sector, Blockchain can optimize the management of retail supply chains. It helps track the distribution process, manage inventory levels, control data information, generate accurate financial reports, and even process transactions through smart contracts.
* The interesting aspect is that in each field, Blockchain brings transparency, security, and optimization in managing information and transactions. These are just a few examples, and the potential of Blockchain holds many other possibilities that we can explore and develop in the future.

### Advantages and Disadvantages of Using Blockchain

* Advantages
* Open: One of the major advantages of blockchain technology is that it is accessible to all means anyone can become a participant in the contribution to blockchain technology, one does not require any permission from anybody to join the distributed network.
* Verifiable: Blockchain technology is used to store information in a decentralized manner so everyone can verify the correctness of the information by using zero-knowledge proof through which one party proves the correctness of data to another party without revealing anything about data.
* Permanent: Records or information which is stored using blockchain technology is permanent means one needs not worry about losing the data because duplicate copies are stored at each local node as it is a decentralized network that has a number of trustworthy nodes.
* Free from Censorship: Blockchain technology is considered free from censorship as it does not have control of any single party rather it has the concept of trustworthy nodes for validation and consensus protocols that approve transactions by using smart contracts.
* Tighter Security: Blockchain uses hashing techniques to store each transaction on a block that is connected to each other so it has tighter security. It uses SHA 256 hashing technique for storing transactions.
* Immutability: Data cannot be tampered with in blockchain technology due to its decentralized structure so any change will be reflected in all the nodes so one cannot do fraud here, hence it can be claimed that transactions are tamper-proof.
* Transparency: It makes histories of transactions transparent everywhere all the nodes in the network have a copy of the transaction in the network. If any changes occur in the transaction it is visible to the other nodes.
* Efficiency: Blockchain removes any third-party intervention between transactions and removes the mistake making the system efficient and faster. Settlement is made easier and smooth.
* Cost Reduction: As blockchain needs no third man it reduces the cost for the businesses and gives trust to the other partner.
* Disadvantages
* Scalability: It is one of the biggest drawbacks of blockchain technology as it cannot be scaled due to the fixed size of the block for storing information. The block size is 1 MB due to which it can hold only a couple of transactions on a single block.
* Immaturity: Blockchain is only a couple-year-old technology so people do not have much confidence in it, they are not ready to invest in it yet several applications of blockchain are doing great in different industries but still it needs to win the confidence of even more people to be recognized for its complete utilization.
* Energy Consuming: For verifying any transaction a lot of energy is used so it becomes a problem according to the survey it is considered that 0.3 percent of the world’s electricity had been used by 2018 in the verification of transactions done using blockchain technology.
* Time-Consuming: To add the next block in the chain miners need to compute nonce values many times so this is a time-consuming process and needs to be speed up to be used for industrial purposes.
* Legal Formalities: In some countries, the use of blockchain technology applications is banned like cryptocurrency due to some environmental issues they are not promoting to use blockchain technology in the commercial sector.
* Storage: Blockchain databases are stored on all the nodes of the network creates an issue with the storage, increasing number of transactions will require more storage.
* Regulations: Blockchain faces challenges with some financial institution. Other aspects of technology will be required in order to adopt blockchain in wider aspect.

### Smart contract introduction

Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary’s involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met.

### How smart contract works

Smart contracts work by following simple “if/when…then…” statements that are written into code on a blockchain. A network of computers executes the actions when predetermined conditions have been met and verified. These actions could include releasing funds to the appropriate parties, registering a vehicle, sending notifications, or issuing a ticket. The blockchain is then updated when the transaction is completed. That means the transaction cannot be changed, and only parties who have been granted permission can see the results.

Within a smart contract, there can be as many stipulations as needed to satisfy the participants that the task will be completed satisfactorily. To establish the terms, participants must determine how transactions and their data are represented on the blockchain, agree on the “if/when...then…” rules that govern those transactions, explore all possible exceptions, and define a framework for resolving disputes.

Then the smart contract can be programmed by a developer – although increasingly, organizations that use blockchain for business provide templates, web interfaces, and other online tools to simplify structuring smart contracts.

### Benefits of smart contracts

* Speed, efficiency and accuracy:

Once a condition is met, the contract is executed immediately. Because smart contracts are digital and automated, there’s no paperwork to process and no time spent reconciling errors that often result from manually filling in documents.

* Trust and transparency:

Because there’s no third party involved, and because encrypted records of transactions are shared across participants, there’s no need to question whether information has been altered for personal benefit.

* Security:

Blockchain transaction records are encrypted, which makes them very hard to hack. Moreover, because each record is connected to the previous and subsequent records on a distributed ledger, hackers would have to alter the entire chain to change a single record.

* Savings:

Smart contracts remove the need for intermediaries to handle transactions and, by extension, their associated time delays and fees.

## DAPP (Decentralized Application) overview

### DAPP (Decentralized Application) introduction

dApps or decentralized apps are applications or programs that run on the blockchain. Since we’re talking about “decentralized” apps, they’re free from the control and interference of a single authority. These apps use smart contracts (source code) for their logic.

### Properties of dApps

* Decentralized: dApps operate on decentralized platforms where no one person or group has control. All records of the application’s operation are stored on a public and decentralized blockchain to encourage transparency, trust, and efficiency. Many businesses today are built around the idea of centralization. They are based on a shared database that replicates stored information across all other nodes.
* Incentivized: Anyone that helps secure the application’s blockchain is rewarded with cryptographic tokens.
* Deterministic: dApps are not dependent on the environment in which they get executed, they perform the same functions everywhere.
* Isolated: dApps are isolated from the other transactions and workings in the blockchain. In case there is a bug in the blockchain, the dApps won’t be hampered.
* Open-source: Ideally, all changes are decided by user and developer consensus. Its code base should be open to scrutiny.
* To their contributing users, dApp tokens are generated to provide values to them. Users are granted access to them in exchange for tokens. Miners are also rewarded with tokens when they successfully contribute to the ecosystem.

### Centralized App and Decentralized App Comparison

|  |  |  |
| --- | --- | --- |
| Basic | Centralized Apps | Decentralized Apps |
| Definition | A centralized app is owned by a company and is hosted on a server or servers. For a user to interact with the app, they need to send data back and forth by downloading a copy of the app. The exchange of data is done from the server. | A decentralized app (dApp) is not hosted on any server. It operates on a blockchain. The user can directly engage in transactions without the interference of a central authority. The dApps user will have to pay some amount of cryptocurrency to the developer to use the program’s source code. The source code is also known as the smart contract. |
| Third-Party Involvement | There is a Third-Party Involvement. | There is no Third-Party Involvement |
| Control | The control of the complete application is in the hands of the central authority. | There is no central authority that has control of the application. All the control lies within the app itself. |
| Security | The centralized apps are more prone to hackers and pose a threat to security and data leaks. | The decentralized apps are more prone to hackers and pose a threat to security and data leaks. This is because:  The control isn’t given in the hands of a central authority  There is no single point of failure |
| Ease of Use | The centralized apps are easy to use and provide a good user experience.  It requires the use of a username and password that is easy to remember. | There is no ease of use when it comes to dApps, since there’s no central authority that has the control.  It requires the use of a public and private key to log in, which is not easy to remember. |
| Exchange fees | When it comes to centralized apps, the exchange fees are high. | In decentralized apps, the exchange fees are relatively low. |
| Anonymity | Centralized applications don’t offer anonymity. | The users are anonymous indecentralized applications. |
| Speed | The centralize dapplications are fast. | Dapps can sometimes be slow to load, and payments can take a while to process. |
| Trust | There is no trust in centralized applications because one is taking the word of big corporations, marketing, or PR agencies, for the security and database. These organizations could be corrupt. | In dApps, all of the code is open source so the user can see for themselves what the application does and how it does it. One would never have to take the word of big corporations, marketing, or PR agencies. |
| Downtime | Sometimes, due to a lot of loads, the centralized applications could go down. | Dapps have a zero/low downtime. On the blockchain, it is not possible. |
| Cost | They are cost-effective. | They are costly. |
| Censorship | Centralized applications provide censorship. For example, Twitter censors account if it finds some offensive posts or does it when the government tries to censor accounts if it goes against their agenda. | In the case of decentralization since peers interact directly, there is no or less censorship. |

## Technologies is used in front-end side

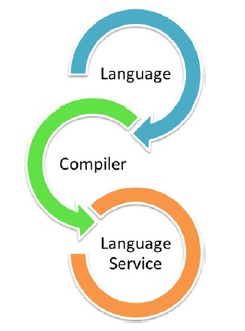
### Typescript definition

By definition, “TypeScript is JavaScript for application-scale development.”

TypeScript is a strongly typed, object oriented, compiled language. It was designed by Anders Hejlsberg (designer of C#) at Microsoft. TypeScript is both a language and a set of tools. TypeScript is a typed superset of JavaScript compiled to JavaScript. In other words, TypeScript is JavaScript plus some additional features.

### Components of TypeScript

* Language − It comprises of the syntax, keywords, and type annotations.
* The TypeScript Compiler − The TypeScript compiler (tsc) converts the instructions written in TypeScript to its JavaScript equivalent.
* The TypeScript Language Service − The "Language Service" exposes an additional layer around the core compiler pipeline that are editor-like applications. The language service supports the common set of a typical editor operations like statement completions, signature help, code formatting and outlining, colorization, etc.



### Why use TypeScript

TypeScript is superior to its other counterparts like CoffeeScript and Dart programming languages in a way that TypeScript is extended JavaScript. In contrast, languages like Dart, CoffeeScript are new languages in themselves and require language-specific execution environment.

### React library introduction

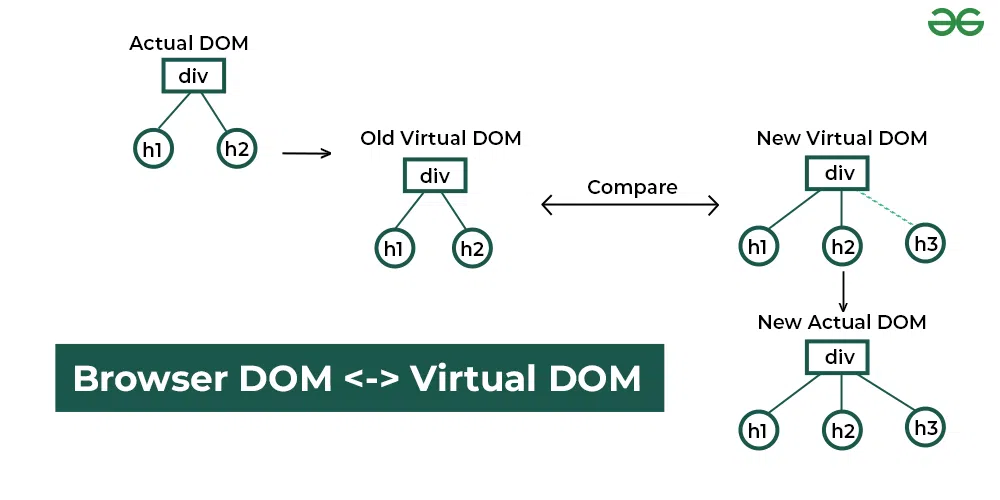
ReactJS is an open-source JavaScript library used to create user interfaces in a declarative and efficient way. It is a component-based front-end library responsible only for the view layer of a Model View Controller(MVC) architecture. React is used to create modular user interfaces and promotes the development of reusable UI components that display dynamic data.

### How does React works?

React creates a virtual DOM in memory to update the browser’s DOM. The virtual DOM will try to find the most efficient way to update the browser’s DOM.

Unlike browser DOM elements, React elements are simple objects and are cheap to create. React DOM takes care of updating the DOM to match the React elements. The reason for this is that JavaScript is very fast and it’s worth keeping a DOM tree in it to speed up its processing.

Although React was designed to be used in the browser, because of its design allows it to be used on the server with Node.js as well.



## Technologies is used in back-end side

### Nodejs introduction

Node.js is an open-source and cross-platform JavaScript runtime environment. It is a popular tool for almost any kind of project!

Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant.

A Node.js app runs in a single process, without creating a new thread for every request. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behavior the exception rather than the norm

When Node.js performs an I/O operation, like reading from the network, accessing a database or the filesystem, instead of blocking the thread and wasting CPU cycles waiting, Node.js will resume the operations when the response comes back.

This allows Node.js to handle thousands of concurrent connections with a single server without introducing the burden of managing thread concurrency, which could be a significant source of bugs.

Node.js has a unique advantage because millions of frontend developers that write JavaScript for the browser are now able to write the server-side code in addition to the client-side code without the need to learn a completely different language.

In Node.js the new ECMAScript standards can be used without problems, as you don't have to wait for all your users to update their browsers - you are in charge of deciding which ECMAScript version to use by changing the Node.js version, and you can also enable specific experimental features by running Node.js with flags.

### Express.js

Express provides a minimal interface to build our applications. It provides us the tools that are required to build our app. It is flexible as there are numerous modules available on npm, which can be directly plugged into Express.

Express was developed by TJ Holowaychuk and is maintained by the Node.js foundation and numerous open source contributors.

## NoSQL and MongoDB overview

### NoSQL definition

When people use the term “NoSQL database”, they typically use it to refer to any non-relational database. Some say the term “NoSQL” stands for “non SQL” while others say it stands for “not only SQL”. Either way, most agree that NoSQL databases are databases that store data in a format other than relational tables.

### Difference between RDBMS and NoSQL databases

While a variety of differences exist between relational database management systems (RDBMS) and NoSQL databases, one of the key differences is the way the data is modeled in the database. In this section, we'll work through an example of modeling the same data in a relational database and a NoSQL database. Then, we'll highlight some of the other key differences between relational databases and NoSQL databases.

### MongoDB Definition

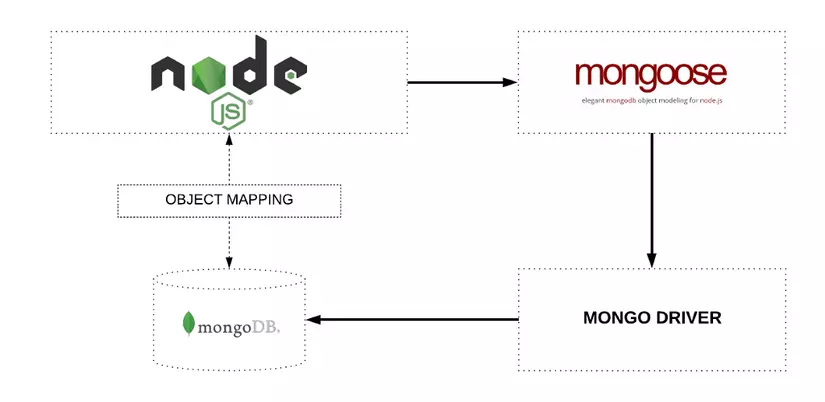
MongoDB, the most popular NoSQL database, is an open-source document-oriented database. The term ‘NoSQL’ means ‘non-relational’. It means that MongoDB isn’t based on the table-like relational database structure but provides an altogether different mechanism for storage and retrieval of data. This format of storage is called BSON ( similar to JSON format).

### Features of MongoDB

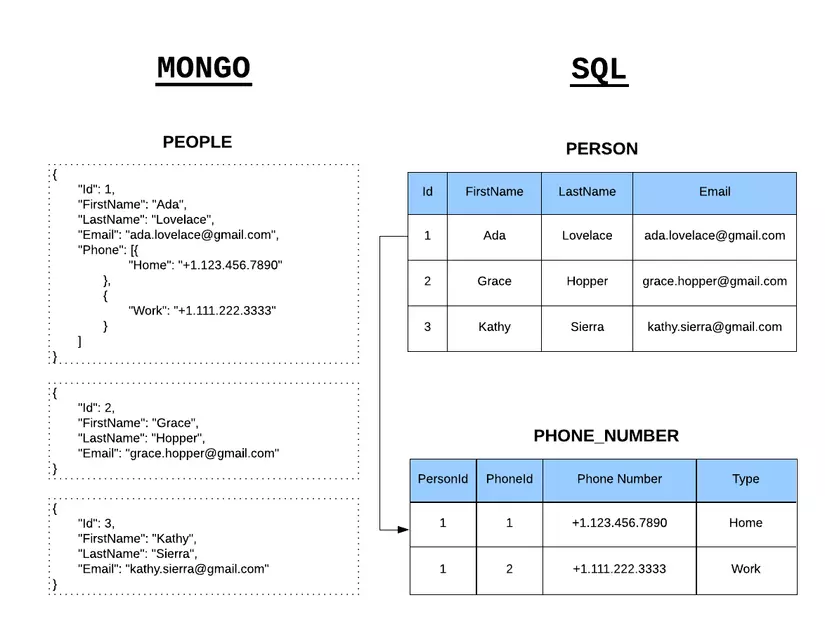
* Document Oriented: MongoDB stores the main subject in the minimal number of documents and not by breaking it up into multiple relational structures like RDBMS. For example, it stores all the information of a computer in a single document called Computer and not in distinct relational structures like CPU, RAM, Hard disk, etc.
* Indexing: Without indexing, a database would have to scan every document of a collection to select those that match the query which would be inefficient. So, for efficient searching Indexing is a must and MongoDB uses it to process huge volumes of data in very less time.
* Scalability: MongoDB scales horizontally using sharding (partitioning data across various servers). Data is partitioned into data chunks using the shard key, and these data chunks are evenly distributed across shards that reside across many physical servers. Also, new machines can be added to a running database.
* Replication and High Availability: MongoDB increases the data availability with multiple copies of data on different servers. By providing redundancy, it protects the database from hardware failures. If one server goes down, the data can be retrieved easily from other active servers which also had the data stored on them.
* Aggregation: Aggregation operations process data records and return the computed results. It is similar to the GROUPBY clause in SQL. A few aggregation expressions are sum, avg, min, max, etc

### Nodejs and MongoDB combination

Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It manages the relationship between data, provides schema validation, and is used to translate between objects in code and their representation in MongoDB.



MongDB is a NoSQL database. JSON documents can be stored in it, and the structure of these documents can change as it is not enforced like in SQL databases. This is one of the advantages of using NoSQL as it speeds up application development and reduces the complexity of deployment.



# SYSTEM ANALYSIS AND DESIGN

## System requirements

### Overview about project system

### Functional requirements

### Non-functional requirements

## UML modeling diagrams

### Use-case diagram

# WEBSITE CONSTRUCTION AND RESULT

## Insert title here

## Insert title here

## Summary

*In this chapter, we have ...............*

CONCLUSIONS AND SUGGESTIONS

##### Conclusions

##### Suggestions

REFERENCES

1. A. Dittmar. “More precise descriptions of temporal relations within task models”. In Interactive Systems: Design, Specification, and Verification, 7th International Workshop DSV-IS, Proceedings, pages 151–168, Limerick,Ireland, 5-6 June 2000.
2. E. G. Sirer and B. N. Bershad, "Using Production Grammars in Software Testing", in Proceedings of the Second Conference on Domain Specific Languages, Austin, Texas, October 3-5,1999.
3. Fewster M., Graham D., (1999), *Software Test Automation – Effective Use of Test Execution Tools, Reading,* MA: Addison – Wesley.
4. Fujiwara S, Bochman G, Khendek F, Amalou M, Ghedasmi A. “Test selection based on finite state models”. IEEE Transactions on Software Engineering 1991; 17(6):591–603.

PUBLICATIONS

APPENDIX

**HƯỚNG DẪN**

Báo cáo ĐATN cung cấp thông tin chi tiết về ĐATN cũng như về tác giả của đồ án. Báo cáo được trình bày trên khổ giấy A4, font chữu Time New Roman, cỡ chữ 13, chế độ giãn dòng là 1.2. Lề trên 2.5 cm, lề dưới 2.5 cm, lề trái 3cm, lề phải 2cm. In 2 mặt. Báo cáo ĐATN được trình bày trong khoảng 40 – 60 trang.

SUPERVIOR’S COMMENT

* Nhận xét của GVHD, SV scan/chụp hình nhận xét của GVHD và đưa vào phần này của báo cáo.

ACKNOWLEDGEMENTS

* Lời cám ơn những người đã giúp đỡ SV hoàn thành đồ án.

STATEMENT OF AUTHORSHIP

* Lời cam kết nội dung đồ án không sao chép của bất kỳ ai.

TABLE OF CONTENTS i

* Mục lục của báo cáo ĐATN

ABBREVIATIONS iii

* Danh mục chữ viết tắt (nếu có)

LIST OF FIGURES vi

* Danh mục hình vẽ (nếu có)

LIST OF TABLES vii

* Danh mục bảng biểu (nếu có)

INTRODUCTION 1

* Nêu lý do chọn đề tài
* Mục tiêu và đối tượng của đề tài
* Cấu trúc của báo cáo ĐATN

CHAPTER 1, 2, 3. .............................3

* Tùy thuộc vào đề tài cụ thể, SV phân chia nội dung các chương sao cho hợp lý.

CONCLUSIONS AND SUGGESTIONS 45

* Tóm tắt các kết quả đạt được của ĐATN
* Đề xuất, hướng phát triển trong tương lai

REFERENCES 46

* Trích tài liệu tham khảo

PUBLICATIONS 48

* Danh mục các công trình đã công bố của các tác giả liên quan đến ĐATN (nếu có)

APPENDIX 49

* Đính kèm các phụ lục (nếu có)