UNIVERSITY OF ECONOMICS

FACULTY OF STATISTICS – COMPUTER SCIENCE

GRADUATION REPORT

Major: Management Information System

Specilization: Information System Administration

BUILDING CRM B2B SYSTEM

Student	: Ta Thi Trang
Class	: 46K21.2
Internship unit	: TMA Solutions Binh Đinh
Mentor	: Nguyen Thanh Cong
Instructor	: Master. Cao Thi Nham

Da Nang, May 2024

TABLE OF CONTENT

INTERNSHIP UNIT'S FEEDBACK

ACKNOWLEDGEMENTS

To successfully complete this internship and this report, first of all, I would like to express my gratitude to the esteemed lecturers of the Department of Statistics and Computer Science at the University of Economics - UD for their enthusiasm in imparting knowledge during our time at the university. Especially, I would like to thank Ms. Cao Thi Nham, the instructor in charge of guiding this professional report. Thank you for your dedication and willingness to assist and solve problems whenever we encountered difficulties.

I would like to extend my thanks to TMA Solutions Binh Dinh for providing me with the opportunity to intern here. I appreciate all the colleagues in the department and the company who shared their knowledge and soft skills, as well as helped overcome obstacles during the internship. Special thanks to Mr. Nguyen Thanh Cong, my direct mentor at TMA Binh Dinh, who closely monitored, provided feedback, and supported me significantly throughout the internship.

This professional internship report may have some shortcomings due to the limited time, experience, and abilities of an intern. I sincerely hope to receive feedback and contributions to further improve myself in the future.

Thank you sincerely!

AFFIRMATION

I solemnly affirm that the report on the topic “CRM B2B” represents the outcome of my research during the internship at TMA Solutions Binh Dinh, under the enthusiastic guidance and support of my mentors in DG8-Ext and my supervising instructor, Ms. Cao Thi Nham.

The content of this report reflects my diligent efforts in understanding, researching, and learning throughout my internship at the company. I also acknowledge that I have drawn insights from the materials provided by the company. I take full responsibility for any issues that may arise.

Student’s full name

Ta Thi Trang

IMAGE LIST

TABLE OF CHARTS

ABBREVIATION LIST

INTRODUCTION

1. **Reason for choosing the topic**
2. **Objectives**
3. **Research methodology**
4. **Subjects and research scope**
5. **Content**

The topic is organized into the following sections:

- Introduction
- **Chapter 1:** Overview of intership unit and job position
- **Chapter 2:** Theoretical foundations
- **Chapter 3:** Analyze and design system
- **Chapter 4:** Develop system
- Conclusion and future directions

CHAPTER 1: OVERVIEW OF INTERNSHIP UNIT AND JOB POSITION

1.1. TMA Solutions Binh Dinh

Image 1 - 1: TMA Tech Group's logo

TMA Tech Group, established in 1997, is a leading technology conglomerate in Vietnam with 4,000 engineers. Our clients include top global technology corporations from 30 countries. TMA currently operates 7 branches in Vietnam (6 in Ho Chi Minh City and 1 in Quy Nhon) and 6 international branches (in the United States, Australia, Canada, Germany, Japan, and Singapore).

In June 2018, TMA opened a branch in Binh Dinh. Over the past 5 years, TMA Binh Dinh has experienced rapid growth, employing over 600 engineers, including many who returned to their hometown after working in Ho Chi Minh City.

In August 2018, TMA commenced the construction of the TMA Innovation Park (TIP) on a 10-hectare site in the Quy Nhon Innovation Valley (QNIVY) with an investment of hundreds of billion VND. As the first software center in the Quy Nhon Innovation Valley, TMA Innovation Park aims to become a leading software and high-tech development center in the Central region, contributing significantly to establishing Quy Nhon as a destination for Industry 4.0 technology in Vietnam. TMA Innovation Park includes the Software Development Center, Software Workshop,

R&D Center, Data Science Center, and Technology Academy, covering a total usable area of over 15,000 square meters.

Image 1 - 2: TMA Solutions BinhDinh's logo

Through close relationships and strategic cooperation programs with major universities in the Central Highlands region, such as Quy Nhon University, Tay Nguyen University, Phu Yen University, and Pham Van Dong University, TMA Innovation Park provides a modern working environment that meets international standards. With a workforce of over 3,000 engineers, it serves not only as a place for Central Vietnamese students to start their careers but also as a hub for talent from the Central region nationwide. TMA Innovation Park aims to drive high-tech development, scientific and technical education, and socio-economic growth in Binh Dinh and other Central provinces.

With robust development over the past 25 years, TMA takes pride in earning the trust of clients from large corporations across 30 countries worldwide. TMA has consistently been recognized as a top IT company in Vietnam and has received accolades in various categories, including being among the top 10 software exporters, top 10 Fintech companies, and top 10 AI-IoT companies.

Placing people at the forefront, TMA fosters a professional and friendly working environment. We continuously strive to create a

culturally rich and vibrant atmosphere, ensuring that TMA employees feel comfortable and view the company not only as a workplace but also as a hub for enjoyable and exciting recreational activities.

Additionally, TMA offers attractive benefits, including:

- Competitive salaries and bonuses
- Abundant opportunities for career advancement
- Comprehensive health care insurance with high coverage
- An internal training system with hundreds of courses annually
- Numerous opportunities for international assignments each year.

1.2. Java Developer

1.2.1. What does a Java Developer do?

Java Backend programming is the process of using the Java programming language to build software that handles the logic, data processing, and resource management functions of an application or website. Java is one of the popular programming languages used in Backend programming.

Java Backend applications can be built on platforms such as Spring Framework, JavaServer Pages (JSP), Java Servlet and Enterprise JavaBeans (EJBs). Java offers many features and libraries that support the development of Backend Web applications that meet performance, security, and resource management requirements.

1.2.2. Knowledge and skill

a. Foundation skills

- Object-Oriented Programming
- System design and architecture
- JVM virtual machine
- Java building tools
- Web technology
- Code Version Control

b. Code skills

- Web frameworks
- Java Enterprise Edition
- DevOps
- Priciples of SOLID
- Big Data
- Design pattern

1.2.3. Career roadmap

- **From 0-1 year**: Intern Java Developer: participate in small projects, write basic Java source code, and test software.
- **From the first 1-3 years**: Junior Java Developer: write code, debug, participate in code evaluation and write technical documentation.
- **From 3-5 years:** Mid-level Java Developer: engages in more complex and responsible tasks than new industry programmers.
- **From 5-10 years**: Senior Java Developer: has a decisive role in application development.
- **From 10-15 years**: architect Java Developer: responsible for connecting clients as well as software clients, analyzing requirements, building the entire system architecture and taking responsibility for it

CHAPTER 2: THEORETICAL FOUNDATIONS

2.1. Java

Java is a widely used object-oriented programming language and software platform that runs on billions of devices, including notebook computers, mobile devices, gaming consoles, medical devices and many others. The rules and syntax of Java are based on the C and C++ languages.

Image 2 - 1: Java Programming language

One major advantage of developing software with Java is its portability. Once you have written code for a Java program on a notebook computer, it is very easy to move the code to a mobile device. When the language was invented in 1991 by James Gosling of Sun Microsystems (later acquired by Oracle), the primary goal was to be able to "write once, run anywhere."

New and improved software development tools are coming to market at a remarkable pace, displacing incumbent products once thought to be indispensable. In light of this continual turnover, Java's longevity is impressive; more than two decades after its creation, Java is still the most popular language for application software development—developers continue to choose it over languages such as Python, Ruby, PHP, Swift, C++, and others. As a result, Java remains an important requirement for competing in the job market. [1]

2.2. Spring Boot Framework

Spring Boot helps you to create stand-alone, production-grade Spring-based applications that you can run. We take an opinionated view of the Spring platform and third-party libraries, so that you can get started with minimum fuss. Most

Spring Boot applications need very little Spring configuration. You can use Spring Boot to create Java applications that can be started by using java -jar or more traditional war deployments.

Image 2 - 2: Spring Boot framework

Spring Boot's goals are:

- Provide a radically faster and widely accessible getting-started experience for all Spring development.
- Be opinionated out of the box but get out of the way quickly as requirements start to diverge from the defaults.
- Provide a range of non-functional features that are common to large classes of projects (such as embedded servers, security, metrics, health checks, and externalized configuration).
- Absolutely no code generation (when not targeting native image) and no requirement for XML configuration. [2]

2.3. PostgreSQL

PostgreSQL is an object-relational database management system (ORDBMS) based on POSTGRES, Version 4.2, developed at the University of California at Berkeley Computer Science Department. POSTGRES pioneered many concepts that only became available in some commercial database systems much later.

Image 2 -3: PostgreSQL DBMS

PostgreSQL is an open-source descendant of this original Berkeley code. It supports a large part of the SQL standard and offers many modern features:

- complex queries
- foreign keys
- triggers
- updatable views
- transactional integrity
- multiversion concurrency control

Also, PostgreSQL can be extended by the user in many ways, for example by adding new:

- data types
- functions
- operators
- aggregate functions
- index methods
- procedural languages

And because of the liberal license, PostgreSQL can be used, modified, and distributed by anyone free of charge for any purpose, be it private, commercial, or academic. [3]

In database jargon, PostgreSQL uses a client/server model. A PostgreSQL session consists of the following cooperating processes (programs):

- A server process, which manages the database files, accepts connections to the database from client applications, and performs

database actions on behalf of the clients. The database server program is called postgres.

- The user's client (frontend) application that wants to perform database operations. Client applications can be very diverse in nature: a client could be a text-oriented tool, a graphical application, a web server that accesses the database to display web pages, or a specialized database maintenance tool. Some client applications are supplied with the PostgreSQL distribution; most are developed by users.

As is typical of client/server applications, the client and the server can be on different hosts. In that case they communicate over a TCP/IP network connection. You should keep this in mind, because the files that can be accessed on a client machine might not be accessible (or might only be accessible using a different file name) on the database server machine. [4]

2.4. RESTful API

Image 2- 4: RESTful API

a. API definition

An API (the full form is Application Programming Interface) is a contract (or specification) promised by the software which it will honor if other software wants to interact with it for performing business operations.

API allows two or more software applications to talk to each other through a well-defined computing interface. [5]

b. API purpose

The software applications are developed in pieces. To avoid writing a piece of software multiple times in different places, it is written as a reusable component.

The API helps in making the component standard, reusable, easily understood by the users, and abstract. The abstraction helps in exposing only minimum relevant information to other entities and protect the business logic to perform an action.

Making reusable APIs not only benefits the users, but it also makes easy the developer's life as well. The defined scope of API helps in the designing, testing, building, managing, and versioning of the component. [5]

c. What is REST?

REST is an acronym for REpresentational State Transfer and an architectural style for distributed hypermedia systems. Roy Fielding first presented it in 2000 in his famous dissertation. Since then it has become one of the most widely used approaches for building web-based APIs (Application Programming Interfaces).

REST is not a protocol or a standard, it is an architectural style. During the development phase, API developers can implement REST in a variety of ways.

Like the other architectural styles, REST also has its guiding principles and constraints. These principles must be satisfied if a service interface has to be referred to as RESTful.

A Web API (or Web Service) conforming to the REST architectural style is called a REST API (or RESTful API). [6]

d.HTTP Status code

REST APIs use the Status-Line part of an HTTP response message to inform clients of their request's overarching result. RFC 2616 defines the Status-Line syntax as shown below:

HTTP defines these standard status codes that can be used to convey the results of a client's request. The status codes are divided into five categories.

- 1xx: Informational – Communicates transfer protocol-level information.
- 2xx: Success – Indicates that the client's request was accepted successfully.
- 3xx: Redirection – Indicates that the client must take some additional action in order to complete their request.
- 4xx: Client Error – This category of error status codes points the finger at clients.
- 5xx: Server Error – The server takes responsibility for these error status codes. [7]

CHAPTER 3: ANALYZE AND DESIGN SYSTEM

3.1. Analyze system

3.1.1. Workflow

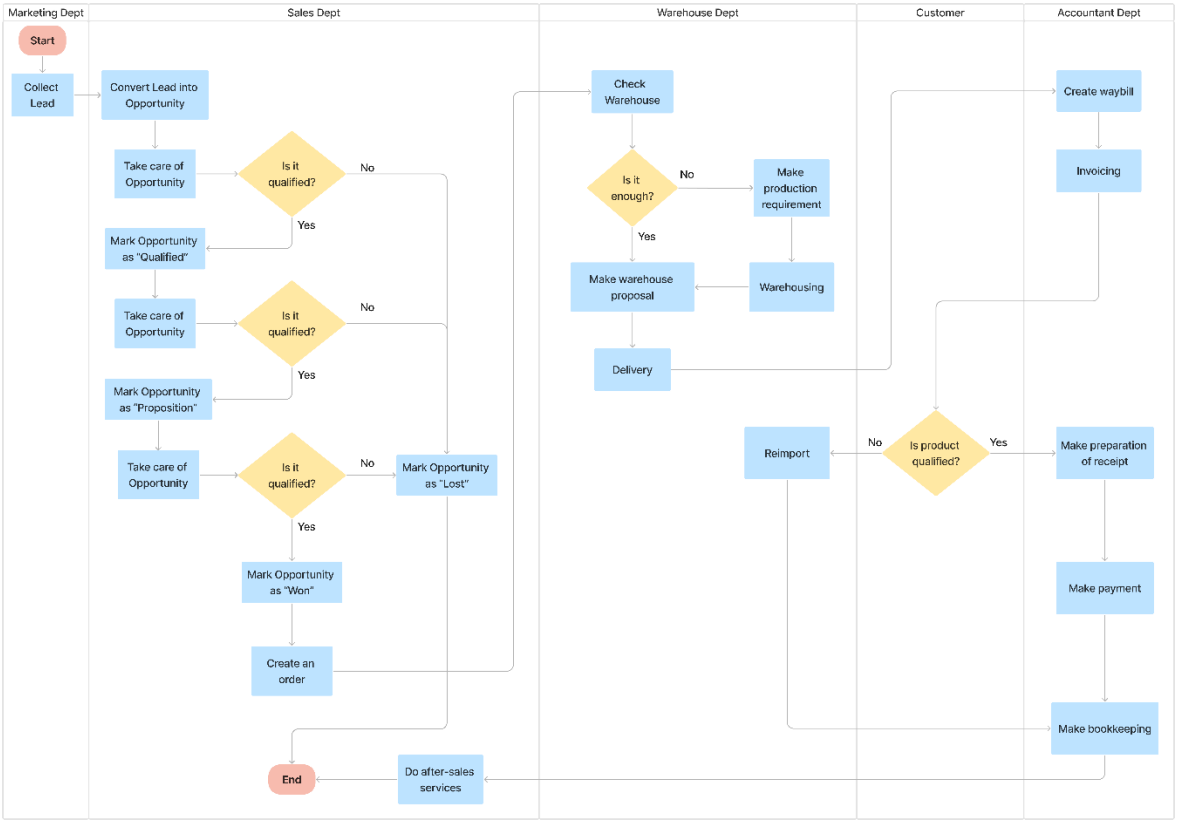

Image 3 – 1: System workflow

3.1.2. Use Case Diagram

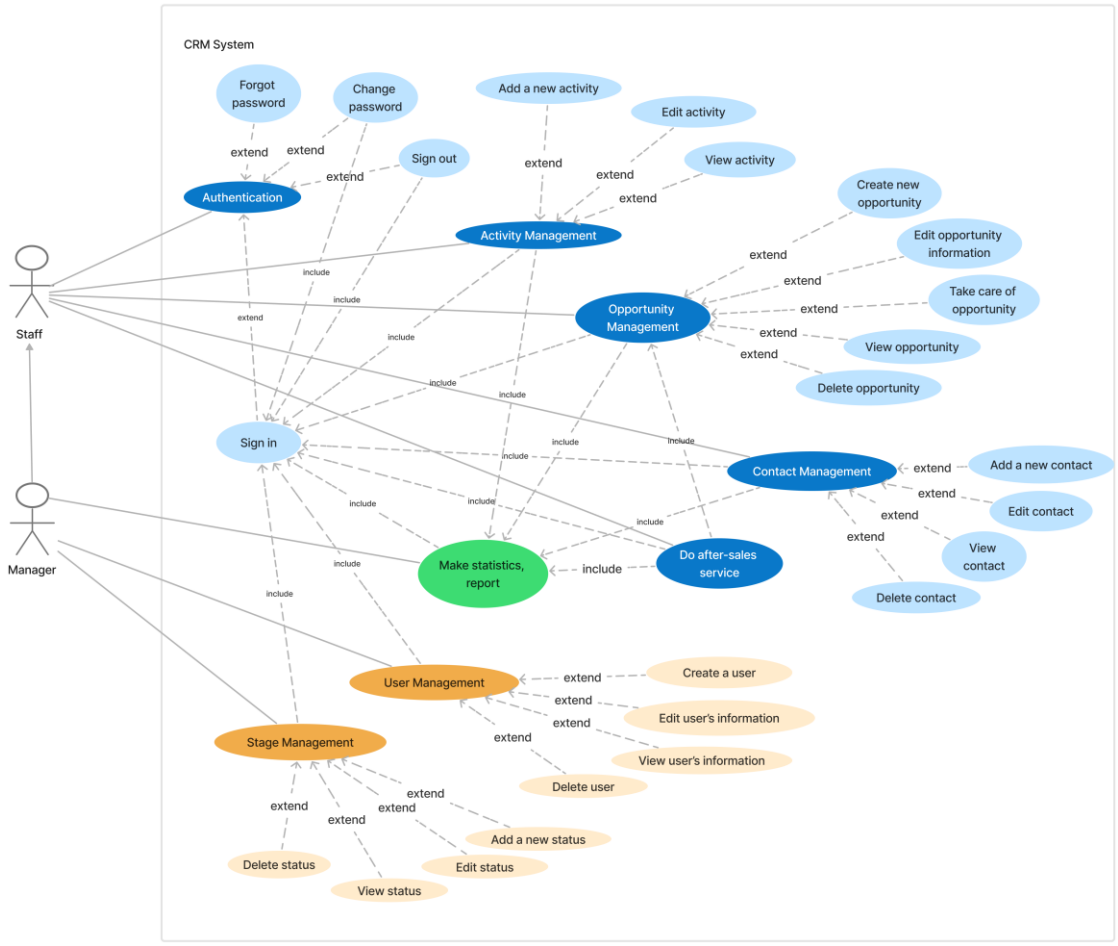

Image 3 – 2: Use case Diagram

3.2. Design system

3.2.1. Entity Relationship Diagram

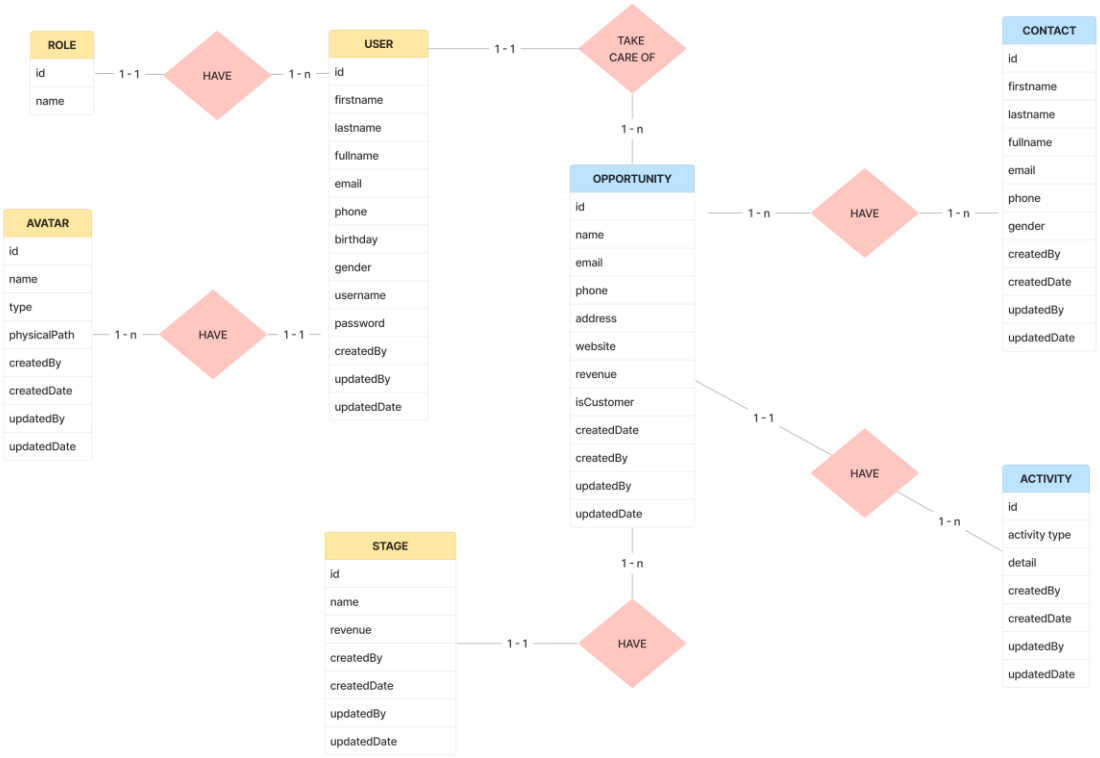

Image 3 – 3: Entity Relationship Diagram

3.2.2.Logical level design

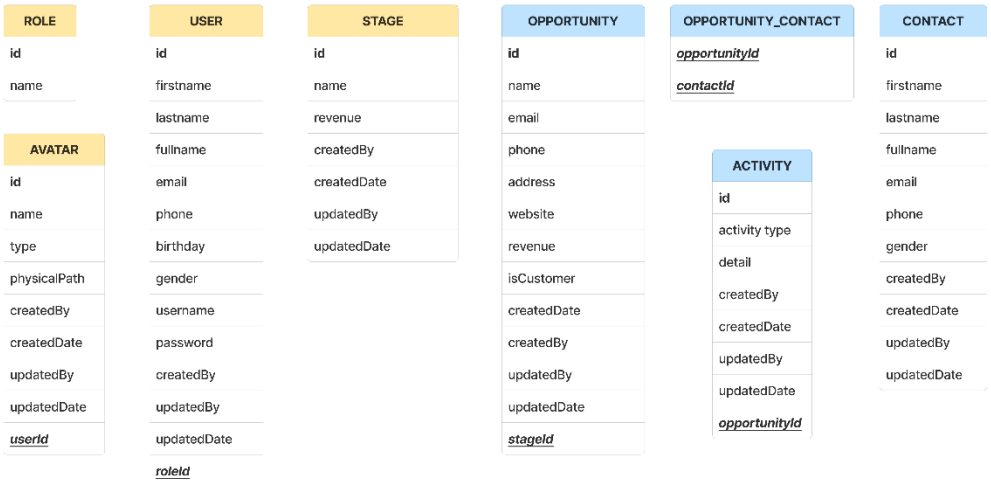

Image 3 – 4: Logical level design

3.2.3. Physical level design

a. User

Key	Attributes	Data type	Constraint	Description
PK	**id**	String	Primary key	User identity
	firstname	String		First name
	lastname	String		Last name
	fullname	String		Full name
	email	String		Email
	phone	String		Phone number
	birthday	Date	YYYY-MM-DD	Birthday
	gender	String		Gender
	username	String	unique, not null	User name
	password	String	not null	Password
	createdBy	String		Person creating's name
	createdDate	Date		Creating date
	updatedBy	String		Person updating's date
	updatedDate	Date		Updating date
FK	***roleId***	Integer	Foreign key	Role identity

Image 3 – 5: Detailed design of User table

b. Role

Key	Attributes	Data type	Constraint	Description
PK	**id**	String	Primary key	Role identity
	name	String		Role name

Image 3 – 6: Detailed design of Role Table

c. Avatar

Key	Attributes	Data type	Constraint	Description
PK	**id**	String	Primary key	Avatar identity
	name	String		Content name
	type	String		File type
	physicalPath	String		Relative path
	createdBy	String		Person creating's name
	createdDate	Date		Creating date
	updatedBy	String		Person updating's name
	updatedDate	Date		Updating date
FK	***userId***	String	Foreign key	User identity

Image 3 – 7: Detailed design of Avatar table

d. Stage

Key	Attributes	Date type	Constraint	Description
PK	**id**	String	Primary key	Stage identity
	name	String		Stage name
	revenue	Double		Total expected revenue
	createdBy	String		Person creating's name
	createdDate	Date		Creating date
	updatedBy	String		Person updating's name
	updatedDate	Date		Updating date

Image 3 – 8: Detailed design of Stage table

e. Opportunity

Key	Attributes	Data type	Constraint	Description
PK	**id**	String	Primary key	Opportunity identity
	name	String		Company name
	email	String		Comany email
	phone	String		Company phone number
	address	String		Company address
	website	String		Company website name
	revenue	Double		Expected revenue
	isCustomer	Boolean	default = False	Is opportunity going to be customer?
	createdBy	String		Person creating's name
	createdDate	Date		Creating date
	updatedBy	String		Person updating's name
	updatedDate	Date		Updating date
FK	***stageId***	String	Foreign key	Stage identity

Image 3 – 9: Detailed design of Opportunity table

f. Contact

Key	Attributes	Data type	Constraint	Description
PK	**id**	String	Primary key	Contact identity
	firstname	String		First name
	lastname	String		Last name
	fullname	String		Full name
	email	String		Email address
	phone	String		Phone number
	birthday	Date		Birthday
	gender	String		Gender
	createdBy	String		Person creating's name
	createdDate	Date		Creating date
	updatedBy	String		Person updating's name
	updatedDate	Date		Updating date

Image 3 – 10: Detailed design of Contact table

g. Opportunity_Contact

Key	Attributes	Data type	Constraint	Description
PK, FK	***opportunityId***	String	Primary key, foreign key	Opportunity identity
PK, FK	***contactId***	String	Primary key, foreign key	Contact identity

Image 3 – 11: Detailed design of Avatar table

h. Activity

Key	Attributes	Date type	Constraint	Description
PK	**id**	String	Primary key	Activity identity
	activity type	String		Manual / Auto
	detail	String		Activity detailed
	createdBy	String		Person creating's name
	createdDate	Date		Creating date
	updatedBy	String		Person updating's name
	updatedDate	Date		Updating name
FK	***opportunityId***	String	Foreign key	Opportunity identity

Image 3 – 12: Detailed design of Activity table

3.3.List API

CHAPTER 4: DEVELOP SYSTEM

CONCLUSION AND FUTURE DIRECTIONS

REFERENCES

[1]: IBM, “What is Java?” [Online]. Available: https://www.ibm.com/topics/java

[2]: docs.spring.io, “Spring Boot Reference Documentation” [Online]. Available: Spring Boot Reference Documentation

[3]: postgresql.org, “What is PostgreSQL?” [Online]. Available: PostgreSQL: Documentation: 16: 1. What Is PostgreSQL?

[4]: postgresql.org, “Architectural Fundamentals” [Online]. Available: PostgreSQL: Documentation: 16: 1.2. Architectural Fundamentals

[5]: restfulapi.net, “What is an API?” [Online]. Available: What is an API? (restfulapi.net)

[6]: resfulapi.net, “What is REST?” [Online]. Available: What is REST?: REST API Tutorial (restfulapi.net)

[7]: restfulapi.net, “HTTP Status Code” [Online]. Available: HTTP Status Codes - REST API Tutorial (restfulapi.net)

QUY ĐỊNH CHUNG

Báo cáo 50 – 80 trang (tính từ phần mở đầu đến hết nội dung, không tính phần tài liệu tham khảo và phụ lục)

Phần Lời mở đầu: 1-5

Phần Giới thiệu / tổng quan: 5 - 10 tr

Phần Lý thuyết: 10 - 20 tr

Phần Triển khai, kết quả, thảo luận…: 30 - 40

Phần Kết luận và hướng phát triển: 1 - 5

Soạn thảo trên trang A4 (trang dọc, lề trái: 3.5cm; trên, phải, dưới: 2.5 cm), Font Times New Roman, canh đều 2 bên, size 13, cách dòng 1.5, cách đoạn trên 6pt, cách đoạn dưới 3pt, hàng đầu tiên lùi vào 1.27 cm; hình và bảng soạn theo caption, chèn trích dẫn chéo (Cross-reference) cho bảng và hình; các danh mục hình, bảng, mục lục làm tự động; đánh số trang như file mẫu (bìa không có số trang, danh mục + mục lục số trang theo i, ii, iii…, nội dung chính theo 1,2,3…)