

[Document title]

[Document subtitle]



[Date]

[Company name]

[Company address]

Table of Contents

[**INTRODUCTION** 2](#_Toc70107562)

[**PART I: OVERVIEW OF TECHNICAL BACKGROUND** 3](#_Toc70107563)

[**CHAPTER 1: BASIC CONCEPTUAL** 3](#_Toc70107564)

[1. Some basic conceptual of web database: 3](#_Toc70107565)

[**CHAPTER 2: NODEJS – THE SERVER-SIDE JAVASCRIPT:** 5](#_Toc70107566)

[1. Operation of Nodejs: 5](#_Toc70107567)

[2. Advantages and Disadvantages of Nodejs: 8](#_Toc70107568)

[**CHAPTER 3: INTRODUCTION ABOUT NOSQL DATABASE:** 9](#_Toc70107569)

[1.Overview about NoSQL: 9](#_Toc70107570)

[2. Document Database: 9](#_Toc70107571)

[3. Operation of NoSQL Database: 9](#_Toc70107572)

[**PART II: ANALYSE AND DESIGN SYSTEM** 11](#_Toc70107573)

[**CHAPTER 1: ANALYSE** 11](#_Toc70107574)

[1. Determine system requirement: 11](#_Toc70107575)

[2. System Requirement: 11](#_Toc70107576)

[**CHAPTER 2: SYSTEM ANALYSIS** 16](#_Toc70107577)

[1. Analysis of Booking Web application: 16](#_Toc70107578)

[2. Database Design: 23](#_Toc70107579)

[3. Activity Diagram: 23](#_Toc70107580)

[CHAPTER 3: USER INTERFACE DESIGN: 24](#_Toc70107581)

[1. Customers interface 24](#_Toc70107582)

[2. Staffs interface 24](#_Toc70107583)

[**PART III: DEPLOYMENT** 25](#_Toc70107584)

[**REFERENCES** 26](#_Toc70107585)

# **INTRODUCTION**

# **PART I: OVERVIEW OF TECHNICAL BACKGROUND**

## **CHAPTER 1: BASIC CONCEPTUAL**

### 1. Some basic conceptual of web database:

#### 1.1. Basic conceptual:

##### 1.1.1. Client/Server Technology:

Client / server technology distributes processing across computers. Essentially, a technology is separated and interpreted by a multitude of devices. Computers as servers are often used to store information that can be accessed from many locations. Servers sit passively waiting to process requests from clients that connect to them. A client sends a request to the server for information or services. The server will take the details and give it to the developer, who will then show it to the customer. Only the unit that executes the Client / Server operation is referred to as the client or server, and the software that runs on this processor is referred to as the client or server program.

##### 1.1.2. Web Server:

Depending on information stored and serving purposes, servers on the internet will be divided into different categories such as web server, email server or FTP Server. Each type of server will be optimized according to the purpose of use.

Web Server is the web that provides information in the form of hypertext, represented in the form of pages. Pages contain links that refer to other pages or to other resources on the same web server, on the same web server.

##### 1.1.3.Database:

A database is a collection of information on a topic, in order to be organized properly in order to conveniently and quickly manage, search and process information.

#### 2.2. Integration between database and web application:

##### 2.2.1. Multi-platform support:

One of the most prominent aspects of the web is that a website can represent a variety of computer families. As a result, in the online database system, clients will have several machines running various operating systems without needing to create software for each machine, but instead instead need to construct a website that is hosted on the web server. This reduces the time and complexity of developing software, as well as the expense of maintaining and upgrading such apps.

##### 2.2.2. Network support:

One important thing to consider when designing database applications is how to access the database from remote computers. As mentioned above, no individual or organization can build each computer network spanning hundreds of thousands of kilometers, while there exists a network with a global scope - the Internet. The new web server and web browser have network capabilities. They are designed to transmit and receive information over the internet or local area network. So why not use the internet for database access purposes.

Web-database integration entails allowing users to view and update data across the internet. To connect the internet with the database, the infrastructure must be rebuilt, including the mechanism for moving information between the internet and the database, the mechanism for accessing the database and complex websites in the web environment, and acts as a database application interface on the Web.

In the Web page, the communication with the user is done through "HTML forms". These forms contain elements used for input such as textbox, checkbox, option button. When the input is completed by clicking a button to accept the submit, this action sends the content of the form to the web server.

Since the Internet is a Web Client / Server system, the database application system on the Web will carry the characteristics of a Client / Server database application system. Therefore, the research on the approach to building Client / Server application system is very necessary. Currently most software developers think that the “component based” approach is a good approach for developing Client / Server applications.

## **CHAPTER 2: NODEJS – THE SERVER-SIDE JAVASCRIPT:**

### 1. Operation of Nodejs:

As previously mentioned, Node.js is a JavaScript runtime environment built on Google's V8 Engine, which allows one to run JavaScript outside of the window. Other aspects you may or may not have heard about Node.js include the fact that it is single-threaded, event-driven, and non-blocking based on the I/O model.

**Nodejs Architecture:** Node.js is developed on the Chrome V8 engine, written in C++, and Libuv, an asynchronous C library that can accommodate Event loop and thread loop events based on I/O synchronization. Node.js can also be used in pure JavaScript, even though it is designed with the V8 engines and Libuvs, both of which are in C and C++ written.

**Nodejs Application:** The Node.js app only runs with a single thread, which makes it possible for only one thread to be used by five users or 5 million users, thereby blocking the Node.js program (which means that a single line of code can block the whole app because an only single thread is being used). Therefore, to ensure that the Node.js program remains operative, asynchronous coding must be used wherever there are callback functions, to avoid asynchronous code from executing and the callback from becoming enforced until the pledge has been solved. In applications, however, synchronous code can be detected, and this is the state before the event-loop is reached. The Event loop allows Node.js programs to conduct non-blocked asynchronous I/o-based operations, that is to say that in this case, all Asynchronous programming is managed and carried out in and beyond the event loop. This means you just want to write synchronous code for transactions that are done once at the start and not always, including to read any data on your computer that can then be requested in asynchronous code by other users (many times).

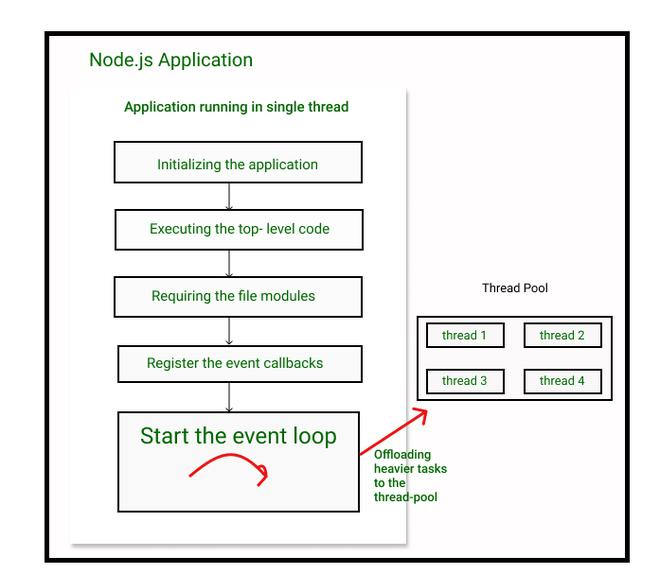


Figure 1: Node application running chart

**The Event-loop:** Much of our asynchronous preparations take place in the event loop. It works on this or these three stages. The whole idea is that:

1. Events will be published. Any asynchronous feature, like getting an HTTP request, a fileSystem module that finished reading a file, or an ending timer, will emit these events. This events are going to change with our code.
2. Event-loop picks them up afterwards.
3. Functions to callback are run.

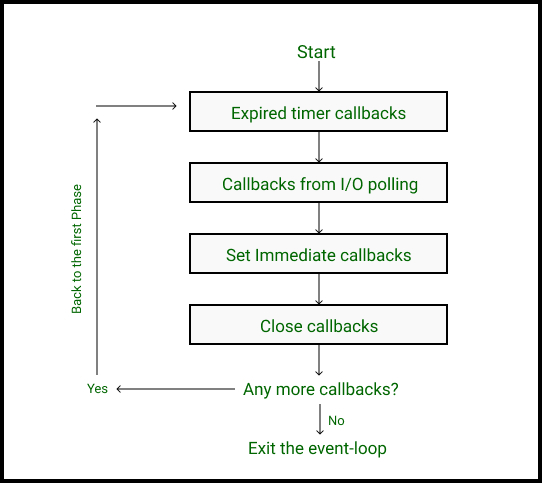


Figure 2: Phases of Event-loop

As the image above illustrated, an event loop has 4 stages, and a callback queue of callbacks from the issued event is set for each process.:

1. The first step consists of Expired timeout calls from the “setTimeout()” module, which are calledback functions.
2. The second step consists of callbacks for I/O polling, such as reading a file or another HTTP request.
3. The “setImmediate()” function callbacks are a callback function the user must do after I/O polling instantly. This feature types may be restricted to specific cases.
4. Closed callbacks are finally posted, such as web server closing, and more.

### 2. Advantages and Disadvantages of Nodejs:

#### 2.1. Advantages:

**Open Source**: Since Node.js is an open source environment, the MIT license makes it free to use. The programming language of Node.js on the framework running on Windows, Mac and Linux. A vast number of open source modules are also supported by Node.js.

**Asynchronous I/O:** Node has been used for asynchronous programming. While no I/O is blocked, you may conduct several operations simultaneously. Users can send several emails easily, carry out databases and read files concurrently.

**Rest API:** Node JS supports RESTful Web Services API creation with great ease.

**Caching Modules:** If the Node JS Modules is loaded for the first time into a program, it is cached. The module code can not be executed more than once in several calls for the load of a module.

**Less Problems with Concurrency:** The Multi-Thread Architecture Node JS does not conform to. It is built on an event loop architecture single-thread. It can easily accommodate any number of requests. Node JS does not have to fix thread list request poooling. Consequently, it is less expensive to handle many requests simultaneously.

**Unit Testing:** JavaScript must write the nodejs application. There are today various platforms and resources available for Java Script Unit Testing, such as the Jasmin Framework.

**Streaming Data:** Streaming-based data applications can be easily developed by using Node JS.

#### 2.2. Disadvantages:

**Inconsistent:** There are some contradictions in the Node JS API. The new API has been upgraded much of the time and programmers must modify the code in order to be consistent with the new API.

**Poorly supported to relational databases**: Relational databases also act oddly when running with js node. If this issue continues, developers in Nodejs can use a No SQL database (mongoDB). The need to learn new databases is eliminated.

## **CHAPTER 3: INTRODUCTION ABOUT NOSQL DATABASE:**

### 1.Overview about NoSQL:

Once the term "NoSQL database" is used, they typically indicate a database that is not related. Some contend that the words "NoSQL" represent "non-SQL." Others argue that they mean "not just SQL." In any event, most people think that NoSQL databases are databases that store information in a different format than relational tables.

A common misconception is that NoSQL databases are not capable of storing relation data, which is also known as non-relational databases. Relation records should be saved in NoSQL databases; they are not stored in relation databases. Many people assume that modeling data with relationship in NoSQL databases is much simpler than in SQL databases, as related data must not be divided among tables.

### 2. Operation of NoSQL Database:

If you look at the SQL and NoSQL database data templates, you can appreciate the appeal from a design point of view in an overimproved example by using address data.

**The SQL Case:** The construction of a SQL database begins with the logical format structure and assumes that the data to be stored remains essentially unmodified. Once the necessary query patterns have been analyzed, a SQL database can improve storage in two tables, a one for specific information and a client, with the last name as the key to both tables.

For eg, every table row is a single client and the following fixed attributes are present in each column:

ID:: name :: age:: address fields :: email address :: phone number

**The NoSQL Case:** The NoSQL database models are available in four types of their own databases as set out in the NoSQL database types section above.

Any type of NoSQL database is designed in accordance with a specific customer condition and for technical purposes each type of database is standardized. The document store is easy to describe since all the basic and client data in a single JSON document will normally be combined. In this example, the column attributes in SQL would be fields, and the customer's record information would be the data values associated with each field.

For example: Property\_title: "Hotel XYZ", Address: "125/8 Nguyen Thi Tan Ward 2 District 8", etc.

### 3. Document Database:

**Document Database:** It comprises JSON – which stand for JavaScript Object Notation and store data in documents. The pairs of fields and values are used in each log. Values which have different forms, such as chains, figures, booleans, arrays or objects; the structures of these values are generally compatible with code objects. Because of the varied field meaning types and effective query languages, document bases are great for a large number of application instances and can be used as a general purpose database. They are able to store large volumes of data horizontally. A common example is MongoDB, regularly classified as the world's famous DB-engine NoSQL database.

**MongoDB:** MongoDB is a document-oriented archive for open source applications. It is used to save a greater number of data and helps you to access these data as well. MongoDB does not depend on the table-like structure of the connection database but offers an entirely different method to save and retrieve data. The word 'non-relational' is here called 'NoSQL.' BSON is the storage format ( similar to JSON format). It uses Storage Engine to handle the way data are stored on the disc and the memory. Multiple search engines are enabled for MongoDB. The default search engine, known as WiredTiger Storage Engine, which is an excellent storage engine, can be used by you and it operates effectively on your results, including reading, writing, etc. If you don't want your own search engine to run.

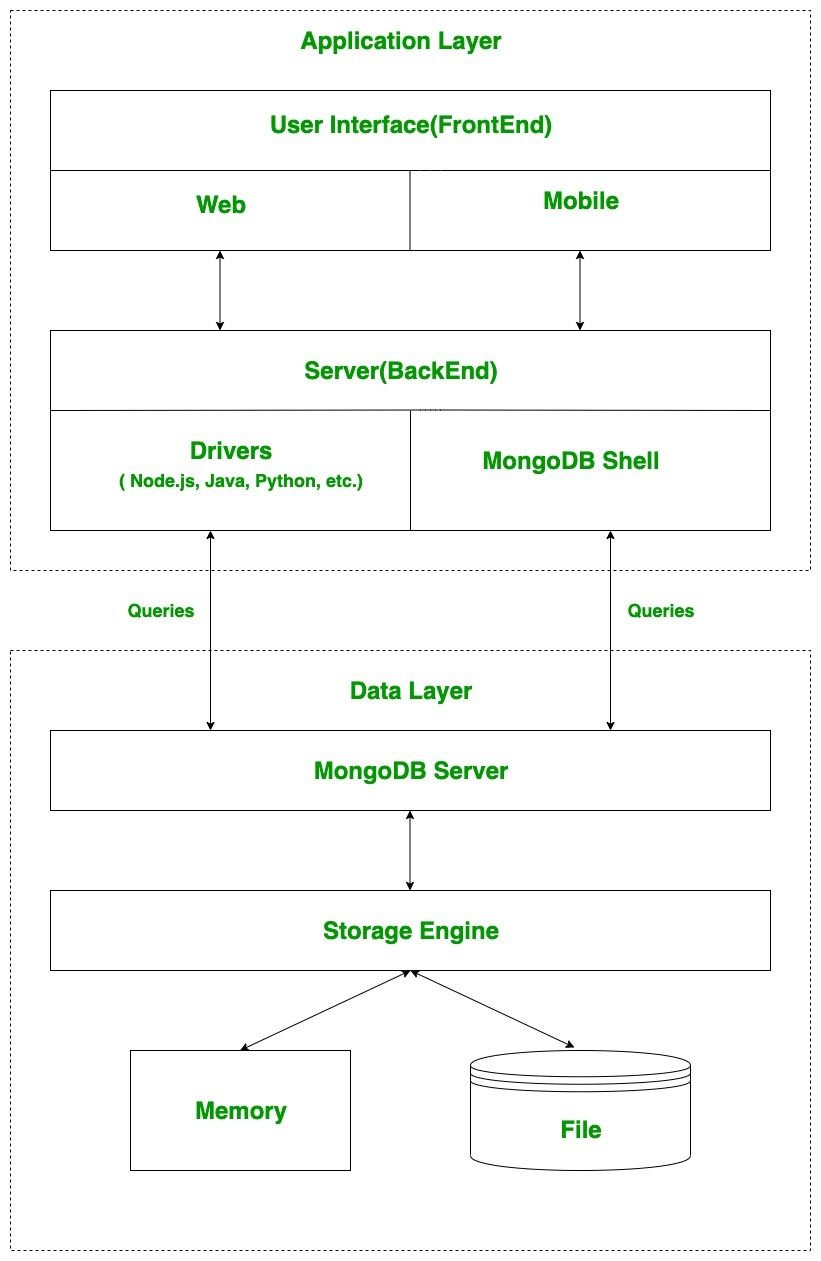


Figure 3: Operation of MongoDB

As the image above illustrated, MongoDb are working in two layer:

**The application layer**: it has two parts, one is a front-end (user interface) and the second is a back-end (server). The interface is the location where users use MongoDB using a web or a mobile device. This site and cell phone includes web sites, mobile apps, android default apps, IOS apps etc. The server involves an interaction between drivers and the MongoDB-Server for queries and a server that is used for logic on the server side.

**Data layer:** These queries are transmitted in the data layer to the MongoDB server. The MongoDB server can now receive the query and pass it to the storage engine. It is not read or write data directly into archives, disk or memory on MongoDB server. The storage engine is responsible for the reading or typing of data in the files or storage, effectively managing the data after transferring the obtained queries to the storage engine.

# **PART II: ANALYSE AND DESIGN SYSTEM**

## **CHAPTER 1: ANALYSE**

### 1. Determine system requirement:

#### 1.1. Core Function:

This is a website that aims to allow users to book and widely recommend hotels to customers as well as provide accurate information. The web app has the following functions:

* Allow to add new hotel information into the database.
* Display a list of hotels by type (including: photos, prices, description ...).
* Display a list of new hotels.
* Allowing registered customers to book hotels.
* Allowing user sign-in with Google account.
* Allow customers add review the hotel.
* Allows customers to pay with visa card or paypal.
* Allow customer to refund within 24 hours.
* Allow customers to view their personal information.
* Allows management of hotels, booking, and customer management.
* Allows management of customer reviews.
* Update hotel, customer.
* Delete hotel,customer (if needed).
* Statistics and tracking the status of customer booking orders.

### 2. System Requirement:

#### 2.1. Use case diagram:

There are two objects using this system, that is, the customers act as bookers and staffs such as admin and manager act as the management of information about the renting hotel service from the partner and deliver it to client. In addition, staffs receive and collect customer reviews and feedbacks to send back to partners and company. For each object, we have the following use cases:

**For customers**: Customers are those who need to find out and book reservations at hotels. They will search the hotel information from the required system from the system and make a reservation at the hotel. So the website must satisfy the following functions:

First: Display a list of hotel information and prices so that the goods can be viewed and selected.

Second: Provide the function to find information about the hotel. With the user's need when entering the service site is searching for the items they need and want. Sometimes there are many customers who visit this website without having the intention to buy or not knowing what to buy, so the system is asking how fast and efficient customers can search for the items they need to find.

Third: After customers have selected the hotel they want, the system must have a payment function so that customers enter the necessary information (such as visa information) and checkout, avoiding ask too much from customers, creating a sense of comfort and privacy for customers.Morover,customer can cancel the reservation when they have issue and refund their money.

In addition, there are a number of functions such as: register, login. Customer can change his password.

As illustrated in the use case diagram, visitors can conduct a hotel search through the web app. After that, they will book a room and make a payment, which will be refunded after 24 hours if there are any issues. At the same time, it is possible to go back to web pages and add hotel’s reviews.

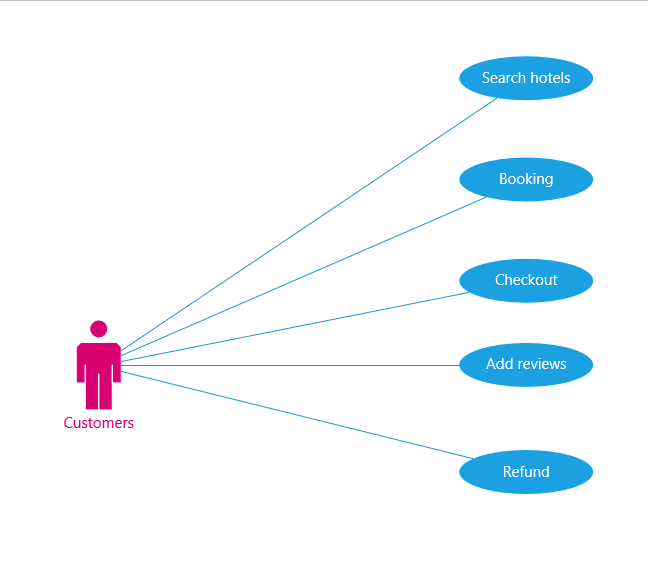


Figure 4: Use case shows interaction between customers with client and customers with server

**For staffs**: As the manager of the application, with the right to control all activities of the system. This person is given a username and password to log into the system to perform their functions.

If the login process is successful, the manager has the following functions:

First: The function of managing updates (adding, deleting, updating) information of hotels on the website.

Second: Receive and check customer reservations,track the status of their orders.

Third: Manage hotel information, orders, and customer information

In addition to the above functions, the website must be presented in a way that is easy to understand, has an easy to use interface, is beautiful, and how customers can see the information to find, provide attractive advertising information leads, to attract interest in your company and have the opportunity to have more people refer.

It is important in a merchant website to ensure absolute safety of user-related information during the checkout process.

At the same time, the website must also be easy to upgrade, maintain, and repair when it is necessary to add and update new features.

As the use case chart shows, staffs can create a new hotel from information provided by partners, can update hotel information if necessary and also delete information about that hotel from Database. Morover, the staff will manage customer reservations when sent to the system as well as customer reviews and feedbacks.

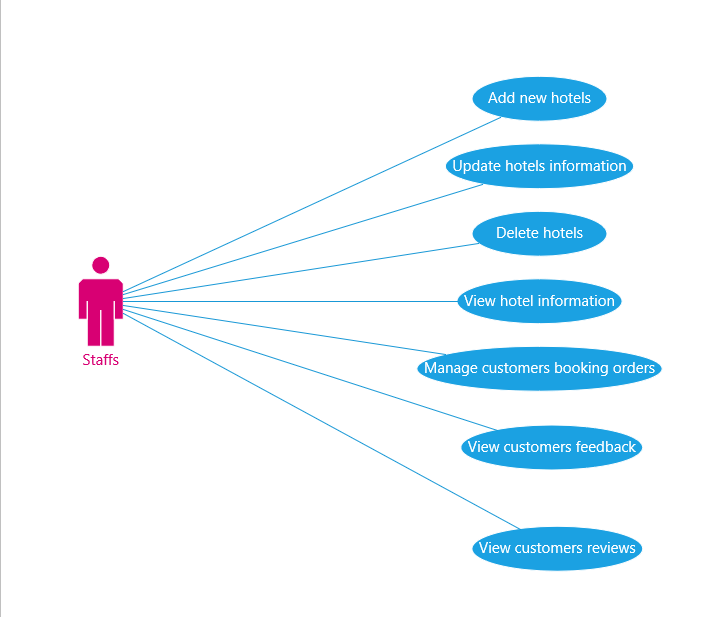


Figure 5: Use case shows interaction between staffs with client and staffs with server

## **CHAPTER 2: SYSTEM ANALYSIS**

### 1. Analysis of Booking Web application:

#### 1.1. Scenario:

##### 1.1.1. Client Scenario:

**Scenario search hotels.**

|  |
| --- |
| Use case: Search hotels. |
| Pre-condition: Customers access to the website home page and enter the hotel information to search and click the Search button |
| User: Customers. |
| Post-condition: Information to be searched for is displayed in the web page. |
| Basic path:   1. Customers access the website homepage. 2. Website appears. 3. Enter the search form 4. Hit the search button 5. The information about the hotel selected by the customer will be sent to the server. 6. The server proceeds to search for the information that the customer chooses in the database. 7. Server returns product information results on the screen of web search.   Use case will end when step 7 executed. |
| Alternative path:  A. When the customer enters the wrong item name in the search field, the information is sent to the server and the response results ask you to re-enter the item to be searched. |

**Scenario booking room.**

|  |
| --- |
| Use case: Customers perform a search for information about the property, make a reservation. |
| Pre-condition: Customers access the website homepage, press the search button, press the button "Booking Now". |
| User: Customers. |
| Post-condition: System will notify payment completed. |
| Basic Path:   1. Customers access the website 2. Website is launched. 3. Customers press the button "read more" to enter the hotel they want to book. 4. Information is sent to the server. 5. Server conducts searching hotel information in the database. 6. Server returns the results to the client. At each hotel there is full information about that hotel (price, description, ...). 7. Customers enter all necessary information such as (date, number of people, room type) and click "Booking Now" button to make a reservation. 8. If the customer has not registered / logged in, the system will go to the page "login". 9. If the customer is a new customer then press the "register" button on the "login" page. 10. Then the “register” page will appear, including full information about the customer, which is required to enter: full name, email, password, re-enter the customer's password to not be wrong. 11. Customer proceeds to enter form information. 12. After entering complete customer information, the customer presses the "register" button to send information to the server. 13. Server gives a success message and the customer continues to log in to continue payment. Then the customer is redirected to the payment page and select the payment method (Visa, Paypal). 14. If the customer agrees to make a reservation, click on the button "checkout". 15. The server receives customer information and gives a website to respond to the customer depending on the payment method that the customer chooses. 16. The customer enters all information requested by the website. 17. Server receives customer input information. 18. Server gives a screen notification that the customer has paid successfully and the invoice will be sent to the customer's email. 19. Then complete information about the order will be sent to the server. 20. Server updates reservation order status to “completed” in the database.   Use case will end when step 20 executed. |
| Alternative Path:  A. At step 9, if the customer is already a member, log in to continue payment.  B. Customers continue to perform from step 13. |

**Scenario cancel and refund.**

|  |
| --- |
| Use case: Customers make cancellation and refund. |
| Pre-condition: Customers press “refund” in “Cart History” page. |
| User: Khách hàng. |
| Post-condition:System notify refund successfully. |
| Basic path:   1. Customers access the website 2. Website is launched. 3. Customers press the button "read more" on the item they want to buy. 4. Information is sent to the server. 5. Server conducts a search for items in the database 6. Server returns the results to the client. At each hotel there is full information about that hotel (price, description, ...). 7. Customers click "Booking Now" button to make the payment. 8. After successful payment, if customer wants to cancel the reservation and refund, then access to "Cart History" page. 9. Click the "Refund" button on the order that the customer wants to cancel and refund 10. Server receives the information and proceeds to cancel the reservation and refund the customer. 11. The server displays a screen notifying that it has successfully canceled and refunded. 12. Server will update the order status as "refunded".   Use case will end when step 12 executed. |
| Alternative path. |

##### 1.1.2. Staff Scenario:

**Scenario add new hotels.**

|  |
| --- |
| Use case: Manager access to the addmin page and enter new product information into the database. |
| Pre-condition: The Manager goes to the index of the admin page and presses the button "add list" |
| User: Manager |
| Post-condition: The newly hotel information is completed and saved in the database. |
| Basic path:   1. Manager access to the website 2. Website sales are launched and there is a section for staff. 3. Server proceed to log into the section for staff. 4. Login page is given, which requires administrator to enter username and password. 5. Manager enter username and password. 6. The Manager proceeds to press the login button to send this message to the server. 7. The server proceeds to check this information. If the information is valid, perform the following steps. 8. Site for staff is launched. In which the website describes the functions of entering, editing, deleting, viewing customer information and website statistics. 9. Manager select "Add list". 10. The web page will show add new form. 11. The Manager proceeds to enter new hotel information. 12. After the import is complete, the Manager proceeds to press the “Save” button to send information about new items to the server. 13. The server performs the crearte new item's content into the database. 14. The server resends the message about successful item entry   Use case will end when step 14 is executed. |
| Alternative path:   * 1. Step 7, the server proceeds to check the staff ’s usename, password.   2. If not valid, the result will be displayed on the screen asking staffs to re-enter usename, password.   3. Staffs performed re-entry, the next steps. |

### 2. Database Design:

**Property Collection:**

### 3. Activity Diagram:

## CHAPTER 3: USER INTERFACE DESIGN:

### 1. Customers interface

### 2. Staffs interface

# **PART III: IMPLEMENTATION**

# **REFERENCES**