VIETNAM NATIONAL UNIVERSITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

GRADUATION THESIS

DEVELOPING A REAL ESTATE SUPPORT SYSTEM: NOSQL DATA CRAWLING AND ANALYZING FOR E-COMMERCE WEBSITES IN VIETNAM

Council: CLC KHMT I

Instructor: Assoc. Prof. Dang Tran Khanh

Reviewer: Assoc. Prof. Tran Minh Quang

Students: Tran Vu Hong Thien (1752506)

Nguyen Thanh Quoc Minh (1752349)

Nguyen Duc Khanh (1752282)

Ho Chi Minh City, July 2021

Acknowledgement

We guarantee that this research is our own, conducted under the supervision and guidance of Assoc. Prof. Dang Tran Khanh. The result of our research is legitimate and has not been published in any forms prior to this. All materials used within this researched are collected by ourselves,by various sources and are appropriately listed in the references section. In addition, within this research, we also used the results of several other authors and organizations. They have all been aptly referenced. In any case of plagiarism, we stand by our actions and are to be responsible for it. Ho Chi Minh City University of Technology therefore are not responsible for any copyright infringements conducted within our research.

Ho Chi Minh City, July 26, 2021

Group of authors

Abstract

Currently in Vietnam, the field of e-commerce is developing extremely strongly. Especially the type of customer-to-customer commerce, typically websites that allow users to sell their own products, is also constantly increasing in size and quality. Along with this development, the need to learn about the market through data sources on e-commerce websites has also increased rapidly. However, in the age of data explosion, when thousands of pieces of information are posted every hour on a huge number of different websites, traditional tools and analytics are gradually becoming overloaded and outdated. Therefore, it is an urgent need to research and build a system that takes care of collecting and analyzing a very large amount of data, replacing the outdated old systems.

In this topic, we research, design and build a system mainly focus on collecting and analyzing real estate data on the e-commerce market in Vietnam. The system includes the main components of data collection, extraction, preprocessing, storage and analysis. Each component is built on technologies suitable for the huge and fast-changing nature of data, typically Big Data technology. Specifically, the work management and distribution mechanisms for multiple processes on multiple machines using the RabbitMQ server; build a generic extractor that can vary depending on the administrator's configuration; Distributed storage and processing of data in various places on Azure Cosmos DB combined with Azure Synapse Analytics. In addition, in this topic, we also solve the problem posed in the data crawling process. It is to solve the problem of anti-crawling that simulates user behavior to prevent the data source from being blocked by the website owner.

Moreover, with the collected data, we also build a recommendation system to assist users in making decisions about the sale of e-commerce real estate by predicting price of the property according to criteria matching the input. The decision support work will be carried out by analysis data according to machine learning models such as Multiple Linear Regression. In addition, the system also integrates the visualization of the data on the collected websites.

To evaluate the system, in this project, we have deployed the whole system on many machines. Then, we run the system from data collected from three different e-commerce websites are chotot.vn, nhadat247.com.vn and batdongsan.com.vn. Measurement data of each component during the system run is collected, and based on that, we

make an assessment of performance, reasonableness as well as remaining limitations, if any. Experimental results show that each component in the system works efficiently and stably. The decision support system also gives accurate and grounded recommendation results, which are also clearly visualized.

The content of the thesis is presented as follows:

- **Chapter 1:** Introduction overview, scope, scientific significance as well as practical significance of the topic.
- **Chapter 2:** Presenting the theoretical bases related to the topic as well as the tools and methods of use.
- **Chapter 3:** Analyze the requirements of the system to be built as well as the functions to be provided.
- **Chapter 4:** Analyze and design the architecture of each component as well as the entire system.
- **Chapter 5:** Describe how to implement the components in the system.
- **Chapter 6:** Presenting the implementation process, experimental data and evaluation.
- **Chapter 7:** Summarize the results achieved as well as the direction of development of the topic.

Contents

1 Introduction **11**

1.1 Problem Statement 11

1.2 Goal 13

1.3 Scope 14

1.4 Scientific significance 14

1.5 Practical Significance 14

2 Methodologies and Theoretical Background **16**

2.1 Related works 16

2.1.1 Data crawling 16

2.1.2 Data extraction problems 18

2.2 Big data analysis 19

2.2.1 Big Data in real estate analysis 20

2.2.2 Modern data warehouse for big data analysis system 21

2.2.2.a Azure Synapse Analytic 22

2.2.2.b Apache Spark 23

2.2.2.c Azure Cosmos DB 26

2.3 Crawling techniques 28

2.3.1 RabbitMQ 28

2.3.1.a Definition 28

2.3.1.b Architecture 28

2.3.1.c The benefits of using RabbitMQ to exchange messages 30
2.3.2 SpaCy 30
2.3.2.a Definition 30
2.3.2.b Architecture 31
2.4 Data Mining 31
2.4.1 Data Mining for Big Data 31
2.4.2 Some data mining techniques 33
2.4.2.a Decision Tree 34
2.4.2.b Regression Methods 35
2.4.2.c Gradient Boosting Machines 37

3 System Requirement Analysis 39

3.1 Real Problem 39
3.1.1 Web Crawling 39
3.1.2 Web Data Extraction 40
3.1.3 Decision making system 40
3.2 Non-Functional Requirement 41
3.2.1 Data Crawling Process 41
3.2.2 Data Parsing Process 42
3.2.3 Data Pre-processing Process 42
3.2.4 Data Storing Process 42
3.2.5 Decision Support System 42
3.3 Functional Requirement 43
3.3.1 Data Crawling Process 43

3.3.2 Data Parsing Process 44
3.3.3 Data Pre-processing Process 44
3.3.4 Data Storing Process 44
3.3.5 Decision Support System 45
3.4 Diagrams 46
3.4.1 Use-case Diagram 46
3.4.1.a Web server of Workers system (Admin) 46
3.4.1.b Normal User (User) 52
3.4.2 Activity Diagram 57
3.4.2.a Administrator of Workers system (Admin) . . . 57
3.4.2.b Normal User (User) 60

4 System Design and Analysis 64

4.1 General architecture 64
4.2 Data crawling component 66
4.3 Data parsing component 70
4.4 Data pre-processing component 73
4.5 Architecture for storing and processing big data of the system 76
4.6 Architecture for recommendation system 80

5 System implementation 83

5.1 Crawling Component 83
5.2 Data Parsing Component 84
5.3 Data Normalizing Component 87
5.4 Storing System Component 90

5.5 Data Visualization Component 94

5.6 Prediction System Component 96

6 System Execution And Evaluation **98**

6.1 System execution 98

6.2 Evaluation 99

6.2.1 Data crawling component 99

6.2.2 Data parsing component 100

6.2.3 Data storage component 100

6.2.4 Decision support system 101

7 Conclusion **102**

7.1 Achieved result 102

7.2 Evaluate the significance of the topic 103

7.3 Future development 103

List of Figures

1.1 Demand of searching for real estate in e-commerce websites in Vietnam . 12
2.1 Work flow of Basic crawling 16
2.2 Benefits of big data analysis in real estate 20
2.3 Building a data warehouse in Microsoft Azure 21
2.4 Azure Synapse Analytics 22
2.5 Azure Synapse Analytics's Services 23
2.6 Apache Spark 24
2.7 Spark's Architecture 24
2.8 Cosmos DB 27
2.9 Exchanges architecture of RabbitMQ. 29
2.10 Spacy NLP piplines 31
2.11 Data mining process 32
2.12 Decision tree for predicting heart disease 34
3.1 Use-case diagram for Web server 46
3.2 Use-case diagram for decision support system 53
3.3 Activity diagram of crawling data. 57
3.4 Activity diagram of parsing data. 58
3.5 Activity diagram of testing parser model. 59
3.6 Activity diagram for recommendation system 60
3.7 Activity diagram for viewing recommended prices 61
3.8 Activity diagram for searching real estate posts 61
3.9 Activity diagram for viewing similar property 62

3.10 Activity diagram for view price fluctuations 63
4.1 General Architecture. 64
4.2 Data Crawling component 66
4.3 URL tree graph 67
4.4 Flowchart of Crawling component 69
4.5 Data Parsing component 70
4.6 Flowchart of Data Parsing component 71
4.7 Architecture of preprocessing component. 73
4.8 Architecture of Azure Synapse Link. 74
4.9 Data Modeling characteristic. 77
4.10 Data warehouse architecture. 78
4.11 Architecture of recommendation system 80
4.12 Architecture of Azure Synapse Analytics 81
5.1 Implement of crawling process. 83
5.2 Implement of parsing process. 84
5.3 Implement of preprocessing component. 87
5.4 Stages of normalizing system. 88
5.5 Implementation of Storing System 90
5.6 Schema of html container. 91
5.7 Schema of parser and updated container. 92
5.8 Schema of final_data container. 93
5.9 Schema of config container. 94
5.10 Request data for visualization. 94
5.11 Example report 1 for data visualization. 95

5.12 Example report 2 for data visualization. 95

5.13 Implementation of training prediction model. 96

5.14 Implementation of training prediction model. 97

6.1 Evaluation of prediction model. 101

List of Tables

3.1 Use case specification: Crawling configuration 47

3.2 Use case specification: Start Crawling 47

3.3 Use case specification: Stop Crawling 48

3.4 Use case specification: Parsing configuration 49

3.5 Use case specification: Start parsing 50

3.6 Use case specification: Stop parsing 50

3.7 Use case specification: Parser Model Configuration 51

3.8 Use case specification: Parser Model Testing 52

3.9 Use case specification: Enter description 54

3.10 Use case specification: View recommended prices 54

3.11 Use case specification: Search real estate posts 55

3.12 Use case specification: View similar property 55

3.13 Use case specification: View price fluctuations 56

5.1 Attributes of Xpath Selector Model 86

5.2 Example for Model filters of "address" attribute of nhadat247.com.vn model 87

1 Introduction

1.1 Problem Statement

Vietnam is one of the fastest growing Internet users in Asia. Along with that trend is the appearance and strong development of E-commerce market. Although it has only appeared recently but Vietnam's E-commerce industry has been growing at an incredible speed. The advent of hundreds of e-commerce websites allowing users to buy and sell products online by businesses, is a typical example of this development. Among the growing types of e-commerce websites in Vietnam, business-to-customer (B2C) websites are up ahead more than other types of remaining e-commerce sites. Most of large e-commerce businesses have their own websites and conveniently allow users to do online shopping. As a result, the websites that allow users to advertise their merchandise, also known as customer-to-customer (C2C), are growing in size and quantity.

Nowadays, more and more people choose the solution of buying and selling real estate on e-commerce sites because of the convenience, variety and ease of making comparisons and references. There are some popular websites that have a huge number of users today such as nha.chotot.vn, alonhadat.com.vn, batdongsan.com.vn, etc.. Vietnam has emerged as a thriving and fast-growing real estate market in Southeast Asia. Suffering from a housing bust in 2009, the market recovered in 2013 and has seen constant and fast growth. Unfortunately, the market was hit hard by the COVID-19 pandemic in 2020 and we saw prices and rents fall dramatically. Through big data analysis on Batdongsan.com.vn at the time of Tet 2021 and Tet 2020, we also see the fact: despite the re-emergence of the Covid-19 epidemic, the demand for real estate search is still increasing.

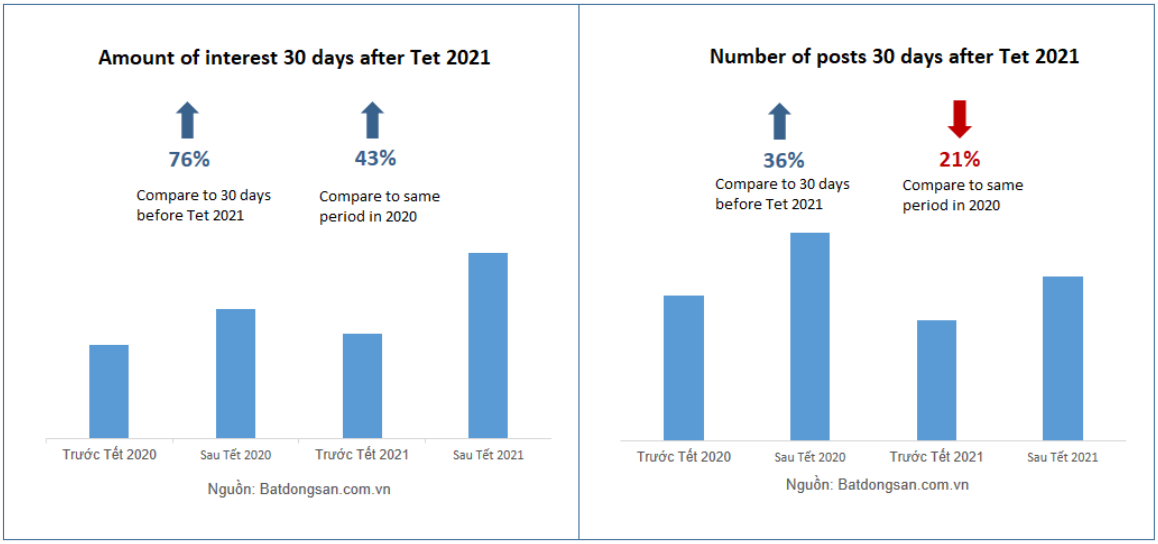


Figure 1.1: Demand of searching for real estate in e-commerce websites in Vietnam

For any business, understand the development trend of the market is one of the keys to success. Especially in the field of C2C, the best way to learn about the market is to collect and do the statistic about the advertising items not only on your website but also on your competitors' websites. Therefore, the construction of the system replaced humans in the collection and processing of data market research is a great demand for e-commerce businesses today. However, building such a system is not simply about developing a software that automatically collects data from websites, but the collected data also needs to be stored and processed. Besides, the collection of data is not always easy, but in fact, this process some of the risks underlying the collection system such as rejected or spoofed data. Since then, there are a number of problems that need to be proposed an appropriate and effective solution as follows:

- **Crawling data from multiple sources**: Getting data from multiple sources helps the system get an objective and general data source from many e-commerce sites so that it can aggregate data from many sources. Make it easier for users to access and synthesize information.
- **Using Big Data technology in processing and storage**: For each website, hundreds of ads are posted every hour, in a variety of formats. Therefore, the data source generated from these websites is increasing in both size, type as well as data update speed. The collection, processing and storage requires to be carried out in a way other than manual methods or applying conventional computer programs, which are only suitable for not too large and complex data.

The e-commerce data collected above is of the nature of big data (Big Data) when it satisfies all three characteristics of the 3V model: volume (volumn) - velocity (velocity) - variety. With the huge nature of volume and complexity as above, big data requires more special techniques for storing, processing and exploiting information than traditional methods. Therefore, it is necessary to apply technologies specifically designed for big data systems. The applied technologies will be presented in detail in the following sections.

- **Developing an application for decision support**: Along with development of a series of C2C websites in Vietnam, the current demand for user support tools is also increasing. When users want to sell their products, especially for ordinary users who do not have much experience in trading, they definitely spend a lot of time learning about the necessary information for them. ads to decide which sites are suitable for listing. This job is relatively difficult and takes a lot of time because users have to compare many similar products on many different websites to make a decision
  Moreover, even if the user has a satisfactory choice, it is not guaranteed to be a "wise" decision. The reason for this, most users are people who do not have much knowledge about the market as well as trading, so it is easy to see that most decisions are made based on their feelings, not necessarily rational ones. . Therefore, the need to build a tool, or a support system to help users make quick and accurate decisions is very urgent. The above decision support system will be based on data collected from various e-commerce websites, namely C2C. This data will be mined to extract useful information for decision making.

1.2 Goal

The main objective of this thesis is to develop an real-estate data collection and analysis system, specifically data from Vietnamese C2C e-commerce websites. The built system needs to meet the data source of the nature of big data, with a distributed collection, storage and processing mechanism. At the same time, the system also needs to solve some problems arising in the collection process such as anti-crawling. In addition to the collected data, the thesis also proposes a decision support application using some machine learning methods for data mining.

1.3 Scope

In the scope of this thesis, we will carry out the following tasks:

- Learn the concepts of e-commerce, B2C, C2C and learn specifically about the Vietnamese e-commerce real-estate market.
- Build a collection of data from C2C websites.
- Building data storage and analysis tools using Big Data technology.
- Build data warehouses to query, analyze, and model with a large amount of information from e-commerce website.
- Develop a decision support application to assist users in not only listing products for sale but also searching for buy on C2C e-commerce websites.

1.4 Scientific significance

- The topic is the synthesis, analysis and experimentation of many techniques related to Big Data processing, as well as data mining methods using machine learning algorithms such as Multiple Linear Regression etc...
- The topic also points out the potential of data sources, not only in the field of e-commerce, but also in other fields, creating a premise to promote research, analysis and utilization of data sources in future.
- Use the Microsoft azure ecosystem to build a big data storage and processing system with the main core of the system being Azure Synapse Analytics - introduced by Microsoft in early 2020.

1.5 Practical Significance

- The topic provides a solution for a complete system for data collection and analysis. Currently, the topic is only encapsulated in the field of real estate e-commerce, but in fact it can be expanded to many other fields such as healthcare, education, etc. in the future, replacing other current technologies are outdated and overloaded.

- The topic also fully researches about market operations as well as how e-commerce websites work and makes a reasonable analysis in setting a direction for an application to collect and collect statistics to help retailers. Businesses have more aggregated information to shape the ongoing market trends and statuses.
- The thesis completes the development of applications to support users in making business decisions in the field of e-commerce.

2 Methodologies and Theoretical Background

2.1 Related works

2.1.1 Data crawling

Web crawler are computer software that travel the whole internet to collects data from a certain website and then synthesizes, analyzes, and extracts information according to the requirements set by the user. Once done, the data will be automatically saved to the database. To put it simply, crawling is using software that helps you get information about a website through the web link you enter, then the software will automatically do the rest to help get the data on the website easier.

There are some important crawling techniques used by Web crawlers, however Focused Web Crawlers is the one we want to mention. This kind of method is also called a sequential crawler. It is initiated by Seeds of URL, then expand the web accessed by using a defined rule. The order of pages visited is determined by frontier data structure. The stop criterion can be the end of list pages have not accessed or reaching the number of needed page or any things.

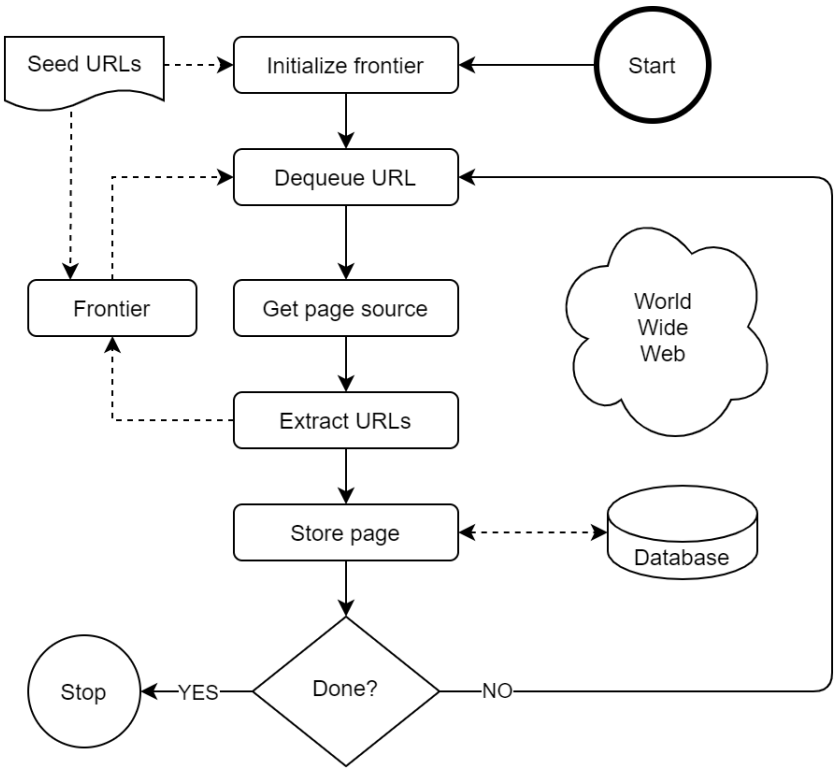


Figure 2.1: Work flow of Basic crawling

The data structure of frontier can be implemented with some algorithms to increase efficiency of crawling:

1. **Breadth First Search (BFS) or Depth First Search (DFS)** - Both are used depend on the rule defined by programmer. BFS is implemented with Queue (FIFO) while DFS is with STACK (LIFO). The two method are simple, but it is possible that may states keep reoccurring. There is no guarantee of finding the goal node.

2. **Best-first search:** - The algorithms can be implemented to search for the best URL in the queue by using heuristic function f(n). The value of the function is evaluated for each URL presents in the queue which has the highest score of priority will be the next visited URL. The algorithm is better than BFS or DFS in find goal but large memory consuming.

3. **Fish-Search:** - The Fish-Search algorithm can be implemented for dynamic search. On one hand, when an URL has relevant information found, it will continue its exploration. On the other hand, it stops. The key principle of the algorithm is as follows: to initiate, it firstly take inputs as a seed URL and a search query. Then it generates list of the next URL from the seeds. At every step, the agent pop an URL, provisionally called parent URL, from queue and fetch the web page. It scores the information using the search query that determined whether the information is relevant or not, the search query return 0-1, base on that score, a heuristic decides whether to pursue the exploration in that direction or not. When the score of parent is high enough, the agent will extracts children URLs, each child will receives predefined depth value, otherwise value will be lower than its parent's depth value. Each of those URL which has the value higher than threshold will be put into the queue and ready for the next step. The algorithm is useful but its limitation here is that there are many URLs have the same priority as it is difficult to assign a more precise scoring function to document which has not been fetched yet.

4. **Generic Algorithm:** - The generic algorithm is an adaptive and heuristic method which is used for improving search result. It exploits several techniques inspired by biological evolution such as inheritance, selection, crossover and mutation. It has four stages. In the first stage, parameters such as population size, generation size, crossover rate or probability, and mutation rate or mutation rate probability are fixed. The original URL is generated by the crawler. On the basis of Jaccard's similar functionality, a fitness value is assigned to the Web site. The higher the fitness

value, the more similar the page is to the domain vocabulary. Jaccard's function, based on links, is the ratio between the number of intersecting links and the merge link between two Web pages. The more common links, the higher Jaccard's score. After the fitness values are calculated, the pages with the better fitness values are selected by a random number generator. Some relevant pages are selected and the rest are discarded. Then all the outbound links are extracted from the leftover pages and cross work to select the most promising URLs. The cross-section of a URL is the sum of the coverage of all the pages that contain the URL. Based on cross value, URLs are sorted and put into crawl queue. Mutation is intended to provide crawlers with the ability to discover many relevant Web pages. Random keywords from the vocabulary are extracted and run as queries in well-known search engines. Top results from search engines and results from cross-stage are combined for more optimal results.

2.1.2 Data extraction problems

There are many techniques which are used to extract data from HTML pages. HTML pages mostly contain semi-structured or unstructured data in which information follows a nested structure. Following are two types of techniques which are used for extracting data from the web pages.

- **Human copy and paste**: Human copy and paste is the most common and traditional approach to extract information from the web sources. The method is high accurate, but low effectiveness cause time consuming.
- **HTML parser:** is super fast real-time parser for HTML web pages. HTML parser widely used by the developers because of its simple design, processing speed and ability to handle real-world HTML. This fast and robust method is used for text extraction, link extraction, resource extraction, screen scraping, etc. Web Data extraction from the web page is one of the fundamental use- case of the HTML parser.
- **Tree-based technique**: Web sites are usually in a semi-structured form, this semi-structured nature of the website is one of the most exploited features in web data extraction. This feature can be represented as labeled ordered rooted tree, in tree labels are used to represent the tags of HTML syntax and the tree represents the different level of nesting of elements constructing the web pages. This representation

of tag and element is known as DOM (Document Object Model). Some techniques are: Addressing elements in the document tree (XPath), Tree edit distance matching algorithms and so on.

- **Web Wrapper:** Web wrapper is a procedure of extracting structured data from unstructured(or semi-structured) data sources that referred as a wrapper. It is a process of implementing one or more classes of algorithm, that finds data based on the user requirements, then extracting them from unstructured web source and converting them into structured data. This process can be further expanded by merging and unifying the information or data in a semi automatic or fully automatic manner. Wrapper life-cycle have three steps wrapper generation, wrapper execution and wrapper maintenance. The method includes: Regular expression based approach, Logic-based approach, Machine Learning approaches and so on.

2.2 Big data analysis

Big data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. But it's not the amount of data that's important. It's what organizations do with the data that matters. Big data can be analyzed for insights that lead to better decisions and strategic business moves.

The concept of big data gained momentum in the early 2000s when industry analyst Doug Laney articulated the now-mainstream definition of big data as the three V's:

Volume: Organizations collect data from a variety of sources, including business transactions, smart (IoT) devices, industrial equipment, videos, social media and more. In the past, storing it would have been a problem – but cheaper storage on platforms like data lakes and Hadoop have eased the burden.

Velocity: With the growth in the Internet of Things, data streams in to businesses at an unprecedented speed and must be handled in a timely manner. RFID tags, sensors and smart meters are driving the need to deal with these torrents of data in near-real time.

Variety: Data comes in all types of formats – from structured, numeric data in traditional databases to unstructured text documents, emails, videos, audios, stock ticker data and financial transactions.

2.2.1 Big Data in real estate analysis

Big data, the vast amount of information generated daily in our connected world, is upending business models in industries from finance to tourism. It is also permeating real estate, a multi-pronged, multi-billion-dollar sector known for its reliance on tradition and intuition.

Before big data, many of the decisions made in real estate were mostly based on gut feel and first impressions. Now, data analysis is one of the main factors in today's decision-making process.

Real estate service providers who are focused on delivering bespoke, customer-centric property solutions typically have satisfied customers. Insights from their data can be used to help better meet the needs of their existing customers, the real estate owners, and be used to position themselves as the real estate partner of choice for prospective customers.

The data revolution makes finding data on proximity, real-time traffic estimations, noise levels, areas of late night activity, restaurants, parks, outdoor activities and customer reviews easy to find, and eliminates some of the typical buyer confusion.Big data is transforming the real estate industry into a well-calculated game of information.

Benefits include:

Figure 2.2: Benefits of big data analysis in real estate

2.2.2 Modern data warehouse for big data analysis system

Data is foundation of real estate assessment, the quantity and quality of data will affect the accuracy and precision of assessment directly.In practice, appraisers always have obsession with experience judgment, such as comparison of price, rate of return, return on capital of building project, cost and composition of construction project. The reason is related to marketization of assessment and related industries.

At present, with the development of measure, storage, computer technology, there are more and more abundant collecting methods for real estate assessment data, everybody is a collector and publisher. Traditional attribute data and spatial data become more and more normative, and it's much easier to get the data.

Database is a subject-oriented, compositive and stable data set which can reflect historical changes. And it's used for supporting management decisions. Database provides current and historical data which can support users' decision, and these data are difficult to get from traditional operation database. Database technique is a general name for all kinds of technique and model which can collect operational data into a centralized environment, and provide effective decision making data. All is for get information users needed more and more quickly and convenient, provide support for decision making. Therefore, we pretend to build a data warehouse in real estate analytic.

A data warehouse is a system that aggregates and stores information from a variety of disparate sources within an organization. The goal of a data warehouse is explicitly business-oriented: It is designed to facilitate decision-making by allowing end-users to consolidate and analyze information from different sources.

In this project, we will build a data warehouse using Microsoft Azure:

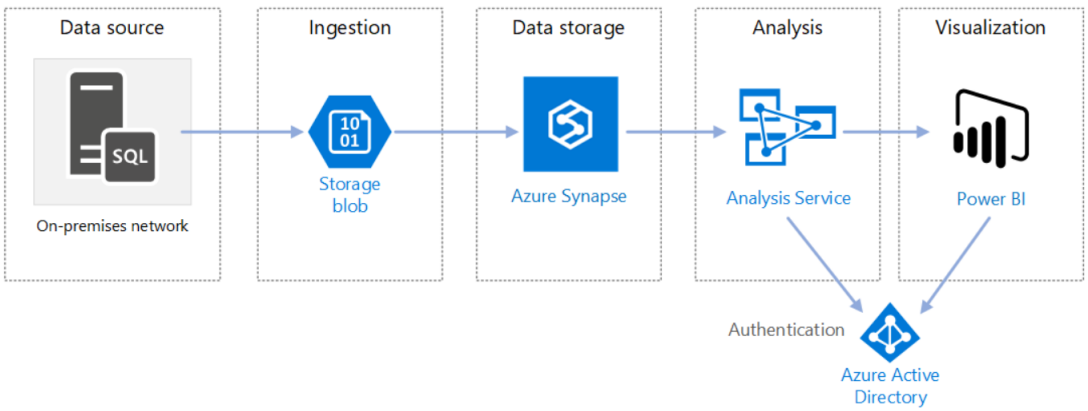


Figure 2.3: Building a data warehouse in Microsoft Azure

In this data warehouse, we use three main components to build: Azure Synapse Analytics, Apache Spark and Azure CosmosDB.

2.2.2.a Azure Synapse Analytic

Azure Synapse is an enterprise analytics service that accelerates time to insight across data warehouses and big data systems. Azure Synapse brings together the best of SQL technologies used in enterprise data warehousing, Spark technologies used for big data, Pipelines for data integration and ETL/ELT, and deep integration with other Azure services.

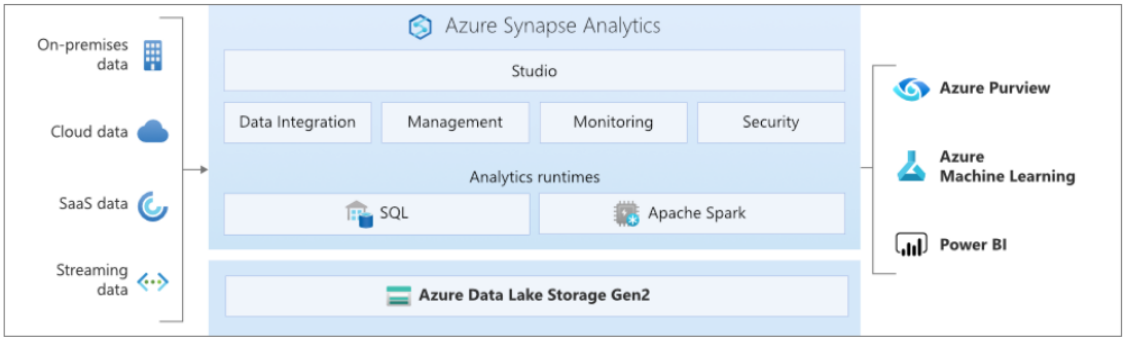


Figure 2.4: Azure Synapse Analytics

Microsoft's service is a SaaS (Software as a Service), and can be used on demand to run only when needed (which has an impact on cost savings). It has four components:

- SQL Analytics with full T-SQL based analysis: SQL Cluster (pay per unit of computation) and SQL on demand (pay per TB processed).
- Apache Spark fully integrated.
- Connectors with multiple data sources.

Azure Synapse uses Azure Data Lake Storage Gen2 as a data warehouse and a consistent data model that incorporates administration, monitoring and metadata management sections. In the security area, it allows you to protect, monitor, and manage your data and analysis solutions, for example using single sign-on and Azure Active Directory integration. Basically, Azure Synapse completes the whole data integration and ETL process and is much more than a normal data warehouse since it includes further stages of the process giving the users the possibility to also create reports and visualizations.

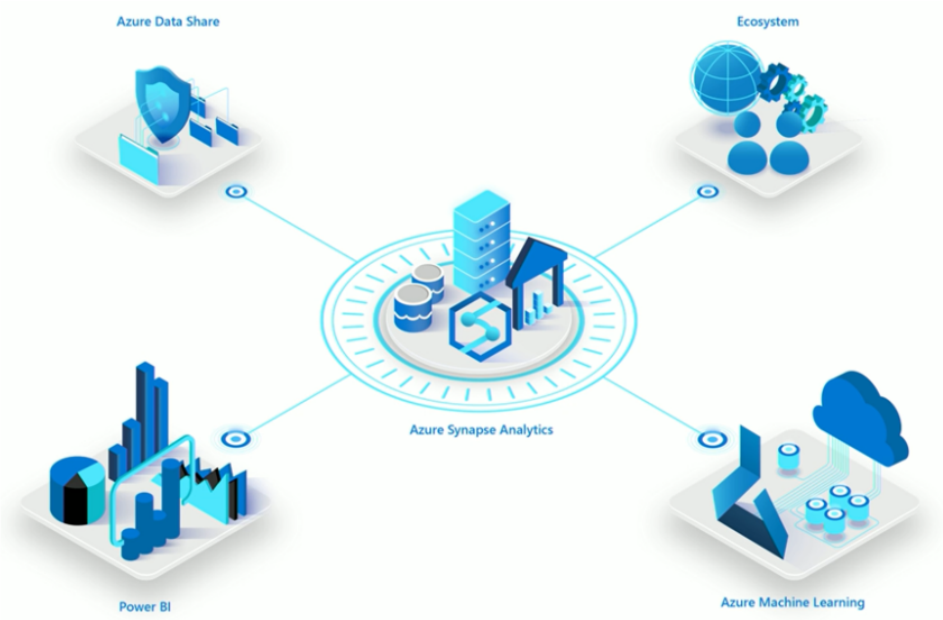


Figure 2.5: Azure Synapse Analytics's Services

Azure Synapse provides you the platform to build and mange a modern DW with limitless analytics service that brings together enterprise data warehousing and Big Data analytics. It gives you the freedom to query data on your terms, using either server less on-demand or provisioned resources at scale.

Real estate assessment is a process of data mining, quantity and quality of the data will impact result of real estate assessment directly. In the past, the problem of lack of samples and false data restricted the accuracy and precision of the results. With the age of big data coming, it solves the problem of data lacking, the new problem is there are too many data. The primary issue is organization, storage and cleaning of massive data. With the development of distributed, cloud technology and data base, the new problem for big data is how to make good use of the data further.Azure Synapse Analytics can solve problems of storage, classification, mining and utilization, lay the foundation for the development of real estate assessment.

2.2.2.b Apache Spark

Apache Spark is a parallel processing framework that supports in-memory processing to boost the performance of big-data analytic applications. Apache Spark in Azure Synapse Analytics is one of Microsoft's implementations of Apache Spark in the cloud. Azure Synapse makes it easy to create and configure a serverless Apache Spark

pool in Azure. Spark pools in Azure Synapse are compatible with Azure Storage and Azure Data Lake Generation 2 Storage.

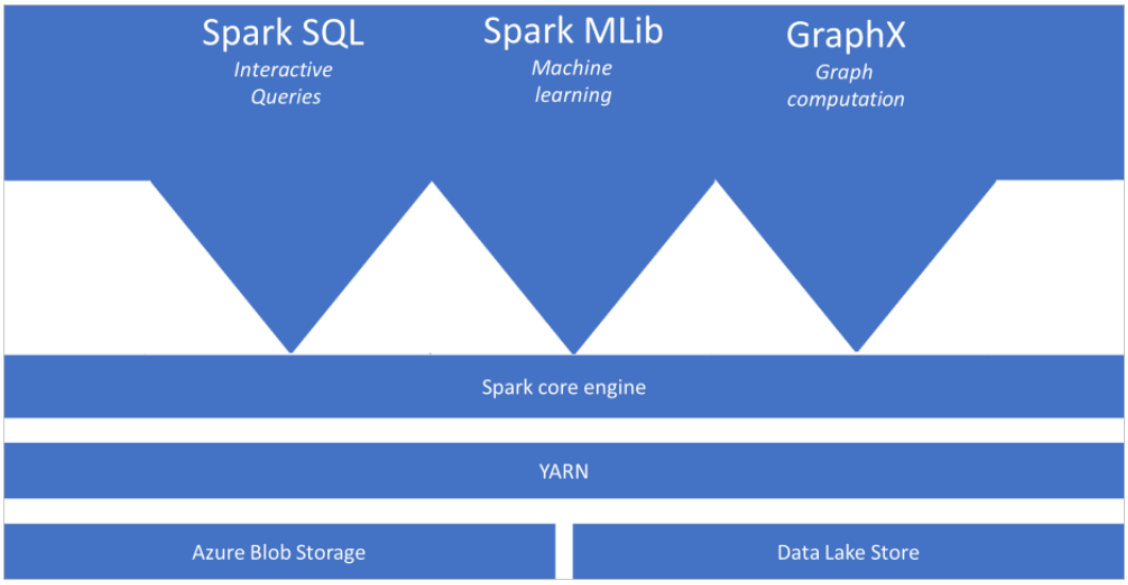


Figure 2.6: Apache Spark

Apache Spark provides primitives for in-memory cluster computing. A Spark job can load and cache data into memory and query it repeatedly. In-memory computing is much faster than disk-based applications. Spark also integrates with multiple programming languages to let you manipulate distributed data sets like local collections. There's no need to structure everything as map and reduce operations.

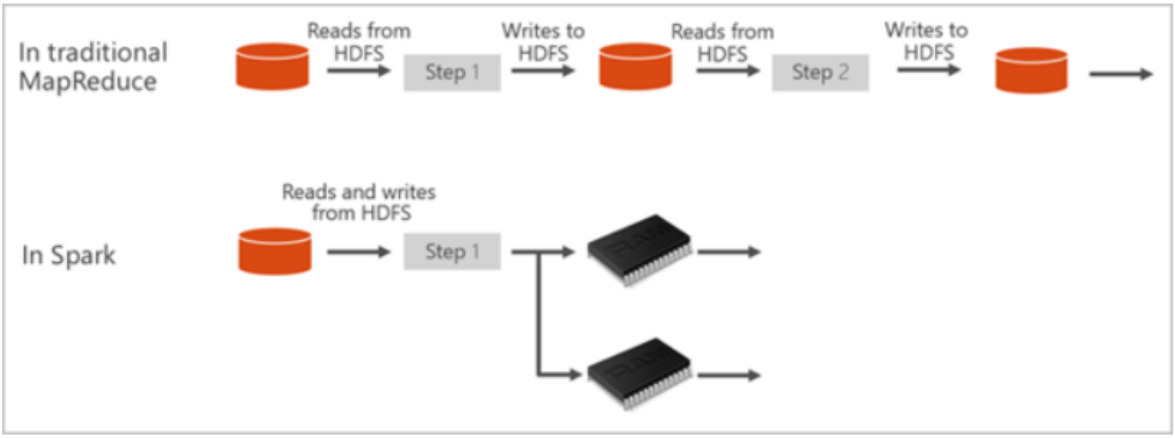


Figure 2.7: Spark's Architecture

Spark pools in Azure Synapse include the following components that are available on the pools by default:

- Spark Core. Includes Spark Core, Spark SQL, GraphX, and MLlib.
- Anaconda. Spark pools in Azure Synapse come with Anaconda libraries pre-installed. Anaconda provides close to 200 libraries for machine learning, data analysis, visualization, etc..
- Apache Livy. Spark in Azure Synapse Analytics includes Apache Livy, a REST API-based Spark job server to remotely submit and monitor jobs.
- Nteract notebook. The nteract ecosystem provides a wide variety of notebook-based applications for your scenario.

Spark pools in Azure Synapse Analytics enable the following key scenarios:

Data Engineering/Data Preparation: Apache Spark includes many language features to support preparation and processing of large volumes of data so that it can be made more valuable and then consumed by other services within Azure Synapse Analytics. This is enabled through multiple languages (C, Scala, PySpark, Spark SQL) and supplied libraries for processing and connectivity.

Machine Learning: Apache Spark comes with MLlib, a machine learning library built on top of Spark that you can use from a Spark pool in Azure Synapse Analytics. Spark pools in Azure Synapse Analytics also include Anaconda, a Python distribution with a variety of packages for data science including machine learning. When combined with built-in support for notebooks, you have an environment for creating machine learning applications.

The price data of real estate information is always the key point and difficult point of data collecting. With the age of big data approaching, many real estate related companies cooperate with each other and share the data they have. The development of internet explored the width and depth of data. Through searching, screening and mining the key word on internet, we can acquire information we could not collect in the past. And then summarize the information, we can extract knowledge from information. The real estate assessment based on knowledge is the development direction in the future. And with Apache Spark in Azure Synapse, we will have a powerful framework supports in-memory processing to boost the performance of big-data analytic applications.

2.2.2.c Azure Cosmos DB

Today's applications are required to be highly responsive and always online. To achieve low latency and high availability, instances of these applications need to be deployed in datacenters that are close to their users. Applications need to respond in real time to large changes in usage at peak hours, store ever increasing volumes of data, and make this data available to users in milliseconds.

Azure Cosmos DB is a fully managed NoSQL database for modern app development. Single-digit millisecond response times, and automatic and instant scalability, guarantee speed at any scale. App development is faster and more productive thanks to turnkey multi region data distribution anywhere in the world, open source APIs and SDKs for popular languages. As a fully managed service, Azure Cosmos DB takes database administration off your hands with automatic management, updates and patching. It also handles capacity management with cost-effective serverless and automatic scaling options that respond to application needs to match capacity with demand.

In this research, we pretend to use the No-SQL Data architecture pattern to store the real estate data. Using the No-SQL database brings more advantages than relational database:

- High throughput: Azure Cosmos DB deployed worldwide across all Azure regions. Partition ranges are capable of being dynamically subdivided to seamlessly grow the database in line with the application, while simultaneously maintaining high availability. Fine-grained multi-tenancy and tightly controlled, cloud-native resource governance facilitates astonishing latency guarantees and predictable performance. Partitioning is fully managed, so administrators need not have to write code or manage partitions.
- Hierarchical data: There are a significant number of use cases where transactions in the database can contain many parent-child relationships. These relationships can grow significantly over time, and prove difficult to manage.If your data contains many parent-child relationships and deep levels of hierarchy, you may want to consider using a NoSQL document database.
- Complex networks and relationships: Azure Cosmos DB is a multi-model database service, which offers an API projection for all the major NoSQL model types; Column-family, Document, Graph, and Key-Value.

- Microservices: The microservices pattern has grown significantly in recent years. This pattern has its roots in Service-Oriented Architecture. The de-facto standard for data transmission in these modern microservices architectures is JSON, which also happens to be the storage medium for the vast majority of document-oriented NoSQL Databases. Azure Cosmos DB in particular has a number of features that make it an even more seamless fit for JSON-based Microservices Architectures than many NoSQL databases.

Following diagram outlines the relation among Cosmos DB account, Database, Collection, and Items:

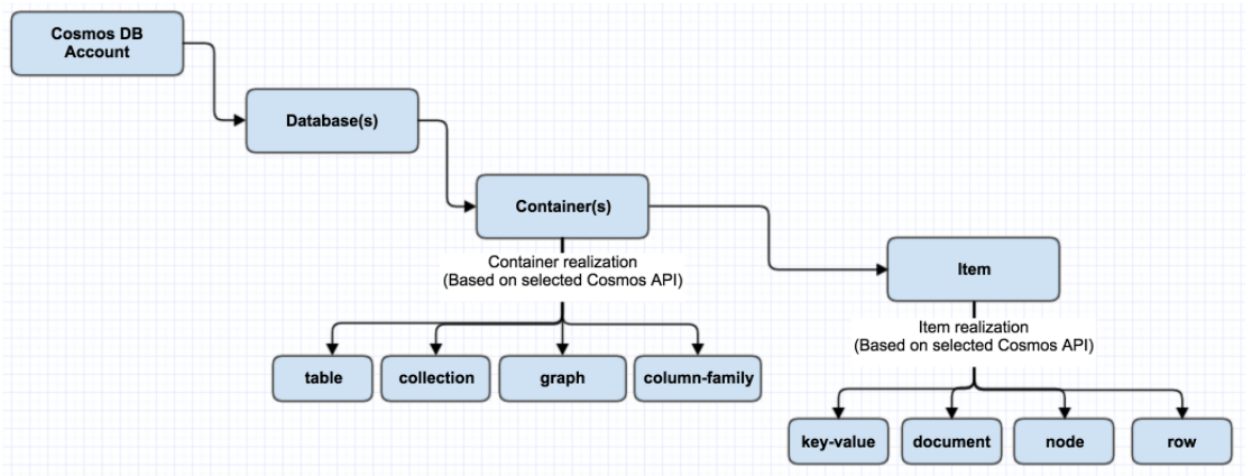


Figure 2.8: Cosmos DB

Azure Cosmos DB Container is schema-agnostic and the unit of scalability. A container is horizontally partitioned and replicated across multiple regions. Based on the partition key, added items in the container and provisioned throughput are distributed automatically across a set of logical partitions. Items in the container don't need to be similar items (e.g., Person, Vehicle, Business Entity, etc.) and they can have arbitrary schemas. Containers can have fixed or unlimited collections. Fixed collections limit you to a single partition, and don't need a partition key set since everything is stored in a single partition. In unlimited, collections are not limited in the number of partitions.
Storage and management of data is the core of entire data base system, it is in charge of interior maintain and management of the data base.The interior maintain of data base include construction of data structure, data manipulation, data maintain and control and data service. The management of data base include data security, file, back up, maintain and recovery. Literature introduced construction of real estate assessment data base de-

tailed.

With Azure CosmosDB, any web, mobile, gaming, and IoT application that needs to handle massive amounts of data, reads, and writes at a global scale with near-real response times for a variety of data will benefit from Cosmos DB's guaranteed high availability, high throughput, low latency, and tunable consistency.

2.3 Crawling techniques

2.3.1 RabbitMQ

Message broker is an intermediary program designed for validating, transforming and routing messages. They serve the communication needs between applications.

With a Message broker, the source application (producer) sends a message to a server process that can provide data orchestration, routing, message translation, persistence, and delivery of all the appropriate destinations (consumer).

There are two basic forms of communication with a Message Broker:

- Publish and Subscribe (Topics)
- Point-to-Point (Queues)

2.3.1.a Definition

RabbitMQ is an AMQP message broker aka message queue management software. Simply put, this is software that defines a queue another application can connect to to drop messages in and send messages based on it.

RabbitMQ helps web servers send responses to requests very quickly instead of being forced to run a resource-hungry procedure on a system. Queuing messages is a good solution when you want to distribute messages to many recipients to reduce the load on workers.

2.3.1.b Architecture

- **Example**

- Online Converter Tool Website, Any user is allowed to convert MP4 video file to MP3 audio file. The problem is that when thousands of users click Convert button, the server receives a lot of requests, which will cause some problems such as slowness, overloaded, can even not respond due to congestion. This is the reason why we need to use RabbitMQ to push these requests into the queue. Hence, the server can process each request, one by one, without getting stuck.

- The mechanism is as follows: A consumer takes messages from the queue and starts processing while a producer is adding new messages to the queue. A request can be made in one language and processed in another. Two applications communicate with each other via messages.

- **Exchanges**

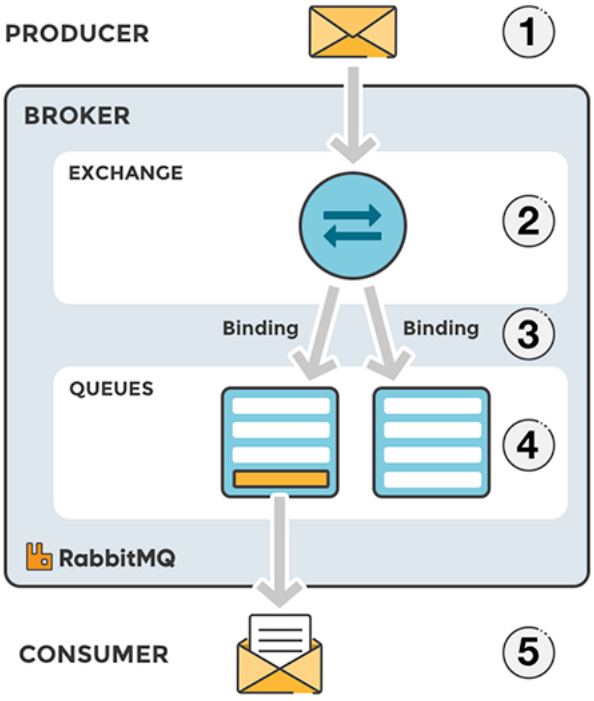


Figure 2.9: Exchanges architecture of RabbitMQ.

- From producer, messages are not directly sent to queue, instead they are put into an exchange. An exchange likes an postman, it delivers messages to indicated queues with the help of bindings and routing keys. Bindings are links between exchange and queues.

- The message flow of RabbitMQ is described as 5 step in the figure above. (1) The producer publish message to an Exchange. (2) Exchange receives message and get its attributes such as routing keys. (3) A binding must be create to send one mes-

sage to queue. The exchange route messages to the queues depending on message attributes. (4) The messages stay in the queue until they are handled by a consumer. (5)The consumer handles the message.

2.3.1.c The benefits of using RabbitMQ to exchange messages

Because the producer sends the message to the consumer through the intermediary message broker So even though the producer and consumer have different languages, the communication process still successful. Currently, RabbitMQ has support for many languages, so it is relatively easy to integrate RabbitMQMQ easy.

One feature of RabbitMQ is asynchronous. Producer doesn't need to know when the message reaches the consumer or when the message is processed by the consumer. The producer's job is to simply send the message to the message broker. The sending of the The message to the consumer in the most reasonable way will be handled by the message broker. Thus, independence between producer and consumer is ensured.

In addition, RabbitMQ has many other interesting features such as:

- Clustering: In RabbitMQ, a cluster is a group of erlang nodes work together. Each erlang node is a machine with a RabbitMQ application operate and share resources: user, vhost, queue, exchange...
- High Availability: Queues can be replicated on multiple machines in the cluster, make sure that even if there are some machines with hardware failure, the messages are still in the ready state.
- High reliability: RabbitMQ provides an ack mechanism that ensures messages are received by consumer has been processed.

2.3.2 SpaCy

2.3.2.a Definition

spaCy is a free and open-source library for Natural Language Processing (NLP) in Python with a lot of in-built capabilities. Unstructured textual data is produced at a large scale, and it's important to process and derive insights from unstructured data. NLP

can help to represent the data in a format that can be understood by computers. spaCy is designed to build information extraction or natural language understanding systems. It's built for production use and provides a concise and user-friendly API.

2.3.2.b Architecture

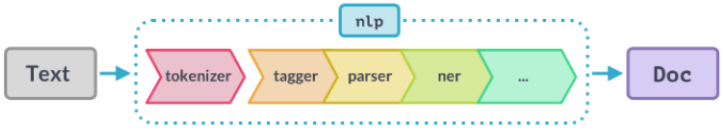


Figure 2.10: Spacy NLP piplines

The processing pipeline consists of one or more pipeline components that are called on the Doc in order. The tokenizer runs before the components. Pipeline components can be added using Language.add_pipe. They can contain a statistical model and trained weights, or only make rule-based modifications to the Doc. spaCy provides a range of built-in components for different language processing tasks and also allows adding custom components.

Named Entity Recognition:

Named Entity Recognition is the process of NLP which has functions of identifying and classifying named entities. Named entities in a text can be classified into persons, organizations, places, money, time, etc. Basically, named entities are identified and segmented into various predefined classes.

NER systems are developed with various linguistic approaches, as well as statistical and machine learning methods. NER has many applications for project or business purposes.

2.4 Data Mining

2.4.1 Data Mining for Big Data

Data mining is a technique for discovering interesting patterns as well as descriptive and understandable models from large-scale data. Data mining can be used to

find correlations or patterns among massive data. It is also the process of discovering or finding some new, valid, understandable, and potentially useful forms of data.

Data mining is a step in the Knowledge Discovery and Data Mining (KDD) process, which entails using data analysis and discovery algorithms to generate a specific enumeration of patterns over the data while staying within appropriate computational efficiency constraints.

The general process of discovering knowledge from data consists of five steps: 1) state the problem, 2) collect the data, 3) pre-process the data, 4) estimate the model , and 5) deploy the model and draw conclusions as shown in Figure 2.10.

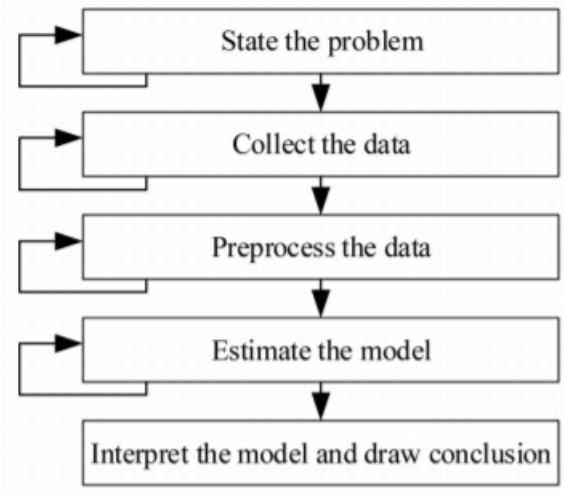


Figure 2.11: Data mining process

Step 1: *State the problem*

It is critical to express the problem of interest issue properly. This step include establishing the mining goals, defining the wished knowledge to extract, data sources and appropriate techniques.

Step 2: *Collect the data*

This stage is about gathering and generating data for the data mining process. Two types of data gathering methods are designed experiments and observational procedures. The data generating process is controlled in the chosen experimental methodology. When using the observational method, data is created at random. Understanding how data is acquired is critical to comprehending the nature of the data that is used. Therefore, a priori knowledge can be advantageous for modeling and interpretation.

Step 3: *Pre-process the data*

After gathering data from existing databases, data warehouses, and data marts, data pre-processing is frequently performed. Pre-processing activities include detecting and scaling outliers, removing missing data, encoding, and choosing features.

- *Outlier Detection and Removal*

  Outliers can be noise resulting from measuring, coding, or recording mistakes, but they may also be crucial signals resulting from hidden unusual occurrences. If the detected outliers are clearly noise, there are two basic strategies for dealing with them: 1) identified and removed, and 2) develop robust model.

- *Scaling, Encoding and Selecting Features*

  Scaling is required to arrange the original data into comparable weights or ranges to be used to the data mining methodology, due to the large volume of data and the many ranges of data accessible today. By providing a reduced number of useful features for data modeling, application-specific encoding methods may be employed to accomplish dimensionality reduction.

Step 4: *Estimate the Model*

The data mining approach is used to design and develop a data mining model at this stage. Developing an acceptable model for tackling the specified problem may be done using a variety of ways. Several possible strategies will be discussed in section 2.4.2.

Step 5: *Interpret the Model and Draw Conclusions*

The outcomes of data mining models must be interpretable and valuable for human comprehension and decision making. A high-dimensional data mining model is supposed to produce very accurate findings.

2.4.2 Some data mining techniques

The two primary kinds of data mining techniques are predictive methods and descriptive methods. In research, health, and engineering, predictive approaches such as decision trees, time series, and regression approaches have become prominent.

In finance, marketing, and database technology, descriptive approaches are frequently utilized. K-means and fuzzy c-means are well-known clustering algorithms in

computer science, engineering, and mathematical modeling. Sequence discovery is often used in medical research to explore links between genes, proteins, and symptoms.

2.4.2.a Decision Tree

Decision Trees are used to assign object membership in different classes based on attributes. The tree structure decision method denotes tests on an attribute with internal or nonleaf nodes. Each branch represents an outcome of a test, and leaf nodes or terminal nodes represent class labels. The definition of a decision tree is as follows:

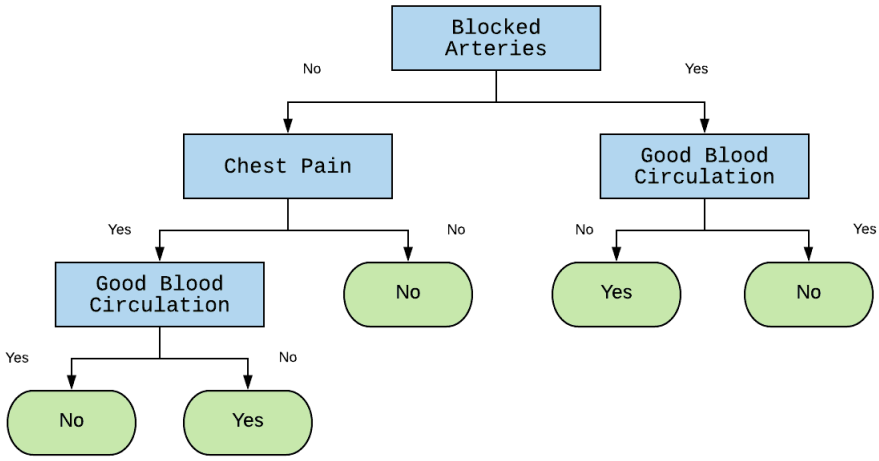


Figure 2.12: Decision tree for predicting heart disease

Suppose that for a database $D = \{t_1,...,t_n\}$, where $t_i = \langle t_{i1},...,t_{ih} \rangle$, the database schema contains the attributes $A = \{A_1,...,A_h\}$ and a set of classes $C = \{C_1,...,C_m\}$. A decision tree or classification tree associated with D has the following properties:

- Each internal node is labeled with an attribute, A_i.
- Each arc is labeled with a predicate that can be applied to the attribute associated with the parent.
- Each leaf node is labeled with a class, C_j.

Solving the classification problem using a decision tree is a two-step process:

- Decision tree induction: construct a decision tree using training data.

- Each arc is labeled with a predicate that can be applied to the attribute associated with the parent.
- For each $t_i \in D$, apply the decision tree to determine its class.

There are several advantages and disadvantage of using a decision tree for classification. For example:

- It is simple to use and efficient, easy to interpret and understand the rules.
- The size of the tree is independent of the size of the database.
- A tree can be built for data that has multiple attributes.
- The rules are simple to grasp and comprehend. As a result, a decision tree can handle a big database.
- A decision tree cannot handle continuous data well.
- The correct branches in the tree cannot be built with missing data and the tree may overfit the training data.
- The decision tree process ignores correlations among attributes.

2.4.2.b Regression Methods

Although alternative approaches exist, regression analysis is a statistical methodology that is most frequently encountered for numerical prediction. Regression also includes identifying distribution trends based on given data. By fitting a set of points to a curve, it may be utilized to address problems like predicting. Linear regression is a simple regression method that depicts the following connection between input and output data:

$$y = c_0 + c_1 x_1 + c_2 x_2 + ... + c_n x_n$$

where c_0,c_1,...,c_n are the regression coefficients and x_0, x_1,..., x_n are the input parameters. y is the output parameter, which depends on the relationship of the input parameters.

Multiple linear regression is employed when there are n input variables, known as the independent variable; one output variable, known as the dependent variable; and $n+1$ constants selected throughout the modeling process to match the input samples. The

relationship between the dependent variable and the independent variable is represented by the following equation:

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon$$

where:

- *y : the predicted value of the dependent variable*
- *B_i : parameter*
- *X_i : independent variable*
- *ϵ : error*

Multiple linear regression can be thought of an extension of simple linear regression, where there are n independent variables, or simple linear regression can be thought of as a special case of multiple linear regression, where *p = 1*. The term 'Linear' is used because in multiple linear regression we assume that *y* is directly related to a linear combination of the independent variables.

As is the case with simple linear regression and correlation, this analysis does not allow us to make causal inferences, but it does allow us to investigate how a set of independent variables is associated with a dependent variable of interest.

Multiple linear regression computes three factors to obtain the best-fit line for each independent variable:

- The regression coefficients that lead to the smallest overall model error.
- The *t-statistic* of the overall model.
- The associated *p-value* (how likely it is that the t-statistic would have occurred by chance if the null hypothesis of no relationship between the independent and dependent variables was true).

It then calculates the *t-statistic* and *p-value* for each regression coefficient in the model. Applications where multiple linear regression may be used include:

- Real estate pricing.
- Predicting the stock market trends.
- Forecasting the number of asthmatics.
- ...

2.4.2.c Gradient Boosting Machines

Gradient boosting machines are a family of powerful machine-learning algorithms that have demonstrated significant effectiveness in a wide range of practical applications. They are extremely adaptable to the specific demands of the application, such as learning with regard to different loss functions. Gradient boosting re-defines boosting as a numerical optimisation problem in which the goal is to reduce the model's loss function by using gradient descent to add weak learners. Gradient descent is a first-order iterative optimisation technique for finding a differentiable function's local minimum. Because gradient boosting is based on minimizing a loss function, there are many different types of loss functions that may be used. Gradient boosting is a versatile approach that may be used to regression, multi-class classification, and other applications since it is based on minimising a loss function.

Gradient boosting, on the surface, appears to be a stage-wise additive approach for generating learners during the learning process (i.e., trees are added one at a time, and existing trees in the model are not changed). The gradient descent optimisation technique is used to determine the contribution of the weak learner to the ensemble. The computed contribution of each tree is predicated on the strong learner's total error being minimized.

Gradient boosting has no effect on the sample distribution since weak learners train on a strong learner's residual mistakes (i.e, pseudo-residuals). This is an alternate method for giving extra weight to misclassified data by training on the model's residuals. Intuitively, new weak learners are being introduced to focus on the areas where the present students are struggling. To minimize the total error of the strong learner, each weak learner's contribution to the final prediction is based on a gradient optimisation procedure.

Gradient boosting consists of three components:

- A loss function that must be optimized.
- The base (or weak) learners to make predictions.
- An additive model to add base learners to minimize the loss function.

Loss Function

The type of loss function applied is determined on the issue being addressed. It must be differentiable, but there are numerous common loss functions available, and also the ability to define your own. For example, in regression, a squared error may be used,

while in classification, a logarithmic loss may be used.

Base-Learner

Different base-learner models can be included into a specific GBM. The base-learner models that are most commonly used in practice will be briefly described and shown. There are three types of base-learner models that are widely used: linear models, smooth models, and decision trees. Other models, such as "Markov random fields" or "Wavelets", are also available, although their use is limited to a few practical applications. The following is the systematization of the base-learner model with the appropriate function examples:

- Linear models: Ordinary linear regression, Ridge penalized linear regression, Random effects.
- Smooth models: P-splines, Radial basis functions.
- Decision trees: Decision tree stumps, Decision trees with arbitrary interaction depth.
- Other models: Markov Random Fields, Wavelets, Custom base-learner functions.

Additive Model

Existing trees in the model are not altered, and new trees are inserted one at a time. When adding trees, a gradient descent technique is utilized to minimize the loss. Gradient descent has traditionally been used to minimize a collection of parameters, such as regression coefficients or weights in a neural network. The weights are adjusted once the mistake or loss has been calculated in order to reduce the error.

Weak learner sub-models, or more precisely decision trees, are used instead of parameters. To conduct the gradient descent process after computing the loss, we must add a tree to the model that decreases the loss (i.e. follow the gradient). This is accomplished by parameterizing the tree, modifying the tree's parameters, and moving in the correct direction by (reducing residual loss.

This method is commonly referred to as functional gradient descent or gradient descent using functions.

The output of the new tree is then combined with the output of the current tree sequence in order to fix or improve the model's final output. When loss reaches a tolerable level or training no longer improves on an external validation dataset, a predetermined number of trees are added or training ends.

Light Gradient Boosted Machine Algorithm (LightGBM)

LightGBM refers to gradient boosting machines that are small and light. It employs a new Gradient-based One-Side Sampling (GOSS) approach to filter out data instances in order to determine a split value. Because of its fast speed, LightGBM is preceded with the word 'Light.' LightGBM is popular because it can manage massive amounts of data while using little memory.

Another reason for LightGBM's popularity is because it places a premium on precision. LGBM also enables GPU learning, thus data scientists are increasingly utilizing it to create data science applications.

It can also handle categorical features by using feature names as input. It is significantly quicker than one-hot coding and does not convert to one-hot coding. To determine the split value of category characteristics, LGBM employs a unique algorithm. Both LighGBM and XGBoost promote leaf growth.

3 System Requirement Analysis

3.1 Real Problem

3.1.1 Web Crawling

Today's website building technology is very diverse and evolving to help the website give the best user experience. Like AJAX or React, for example, websites using this technology make important data invisible to web crawlers while users can see through interaction and use of the browser. More simply, for example, the website chotot.vn, when using a programming library to request a post, will receive the page source with empty content, while the important attributes of the product are located. elsewhere, not in HTML but javascript JSON data. This leads to having to use tools with browser integration to get the content of the website.

Commercial websites use the resources of the server system to serve the access of users. However, if the data collection system makes too many visits to a website at a time and continuously for a long time, their servers will be overloaded and affect the experience of real users. the. That's something the businesses that run those websites are

always looking to fix, because they don't serve crawling bots. Measures they can come up with may be to block persistent access from a computer by blocking IP, requiring chapta code validation when detecting signs of unusual access. Therefore, a system of distributed crawlers will be a potential solution for anti-crawling resistance .

Data collection will be very difficult if the business, owner of the website, knows our collection system is pulling data from their website. Moreover, It is not good if our data collection affects our customers and their business. So the problem is to develop the data collection tool so that it behaves as closely as possible to a user.

3.1.2 Web Data Extraction

The data retrieved by the crawler system is the HTML source code of web pages. This data cannot yet use this resource in data mining. Therefore, it is necessary to design a tool to extract data from HTML sources into data that can be used in analysis and machine learning.

Commercial websites in particular or websites in general today are often very diverse in terms of font-end. As a result, the structure of the source code is very complex and often not the same between websites, even within the same website, but different sections, so the site structure is also different. Besides, because businesses want to limit and make it difficult for data collection tools, they deliberately make the structure of their website's source code often change.

Therefore, the problem is to develop an efficient way to extract data from the HTML source code of different websites.

3.1.3 Decision making system

One of the main needs of users when using C2C e-commerce websites is how to sell products quickly at the most reasonable price and to find products they want as simple and as fast as possible. However, due to the rapid increase in the number of posts on C2C websites as well as the variety of product information including prices, it has created many difficulties when a user wants to post his product. Specifically, difficulty for buyers is finding out quickly which website is selling the product they are looking for, and for sellers, what is the price of their product in the current market.To solve this

need, it is required to develop an application that both helps buyers easily search from many sales pages and supports sellers in decision-making by suggesting the most suitable product price and through other visual ways to know the selling price based on the data that the system crawled and mined.

In addition, the increasingly fierce competition among enterprises has led to the need for research and investment in key products. In the midst of so much data in the e-commerce market, making decisions is extremely difficult and potentially risky. Providing the volatility of products to the business will help a lot for the business operations of the business. Therefore, the application also needs to show the volatility of the product.

3.2 Non-Functional Requirement

3.2.1 Data Crawling Process

- Easy to expand: Adding or removing components do not affect efficiency of whole system.
- Support multi-platforms: Every component is able to run on many different platforms like Windows, Linux,...
- Components send data to sever frequently.
- Perform task properly when received order changing to anti-crawling resistance mode.
- For task management system:
  - Do not delete in queue messages when workers have not finished task.
  - Change task for worker when that worker is informed by e-commerce website that it has been blocked.
  - Respond in time when received message from producer as well as connection request from consumer.
- For worker controller website:
  - User-friendly interface.
  - Web page has fast respond time.
  - Clearly presented, assist admin to make fast and easy configuration.
  - Configuration update must ensure synchronism.

- For anti-crawling:
  - Anti-crawling component need to be attached to worker, it mean having mechanism to support multi-platform and distributing to each worker.

3.2.2 Data Parsing Process

- Ensure data completeness when extracted, not missing.
- Remove special characters.
- Inform when detected important data is NULL.
- For parsing configuration:
  - Has user-friendly interface.
  - Web page has fast respond time.
  - Clearly presented, assist admin to make fast and easy configuration.
  - Configuration update must ensure synchronism.

3.2.3 Data Pre-processing Process

- Data after preprocessing must be meaningfully correct as the original data.
- The data preprocessing must not delete the original data for the purpose of a maintenance mechanism in the event of an error (the error can be caused by the initial data or the preprocessing being wrong).

3.2.4 Data Storing Process

- Easily extensible: Adding or removing components does not affect the architecture of the system.
- Data Availability: Data is available upon request.
- Support fast data query.

3.2.5 Decision Support System

Recommendation system and mobile application need to ensure:

- Friendly and easy to use(user-friendly) user interface.

- Minimize user input operation.
- Providing useful information to support decision-making.
- Fast responding time.
- Data is well visualized.

3.3 Functional Requirement

Based on the analysis of the current status of the components of the system above, the necessary functional requirements that the system needs to ensure are as follows:

- Collect data in parallel and distributed in many places.
- Manage and automatically assign work for workers in the data crawling system.
- Parallel data extraction.
- Support configuring data crawling and extraction.
- The data must be preprocessed before saving to the database.
- Data storage includes many distributed machines in many places, capable of ensuring the data availability, system scalability, distributed data query.
- Capable of supporting decision making, visualizing e-commerce data according to many different aspects.

3.3.1 Data Crawling Process

- The data crawling components are organized into a system and distributed in many places.
- The data crawling process is carried out in parallel and ensures data synchronization data between components.
- There is a mechanism to support anti-crawling, ensuring crawling components is not blocked and the data collected is reliable.
- The collection process is managed using an visual web interface:
  - Allows viewing, deleting, editing configurations for the data collection process.
  - Allows checking the configuration of the data crawling processes is correct or not.

- The collection process is managed using a task management system. The requirements for the task management system are as follows:
  - Automatically assign work to data crawling components.
  - Fair division of work between components: There is no situation where one component has to do too much work while others are in a state of doing nothing.
  - Synchronize data between components: The work divided among components must not overlap, but must be synchronized with each other.
- There are mechanisms to support anti-crawling:
  - Make sure crawling components are not blocked.
  - The collected data is reliable (correct to the real data, not the fake data returned by the web server).

3.3.2 Data Parsing Process

- The data extraction components are organized into a system at the server.
- The data extraction process is performed in parallel and ensures synchronization in updating the database.
- Extract data from HTML source code into computer-processable data.
- The extraction process is configured using an intuitive web interface:
  - Allows viewing, deleting, and editing configurations for data extraction.
  - Allows testing the configuration of data extraction processes.

3.3.3 Data Pre-processing Process

- Data after preprocessing must not contain redundant data (tabs, multiple contiguous spaces, multiple contiguous dots, etc.).
- Standardize number and time formats to a single standard.
- Outliers classification.
- Convert raw data into data that can be processed by computers.

3.3.4 Data Storing Process

- Support data query.

- Data storage must ensure single-digit millisecond response times, and automatic and instant scalability, guarantee speed at any scale.
- Guarantee continuity, 99.999% availability, and enterprise-level security.
- Fully managed and cost-effective.

3.3.5 Decision Support System

Developing a system and mobile application that allow:

- Predict price of real property base on its features.
- Search real estate for sale post on many e-commerce websites.
- Looking for similar property.
- Visualizing price volatility of real estate on many e-commerce websites according to time.

3.4 Diagrams

3.4.1 Use-case Diagram

3.4.1.a Web server of Workers system (Admin)

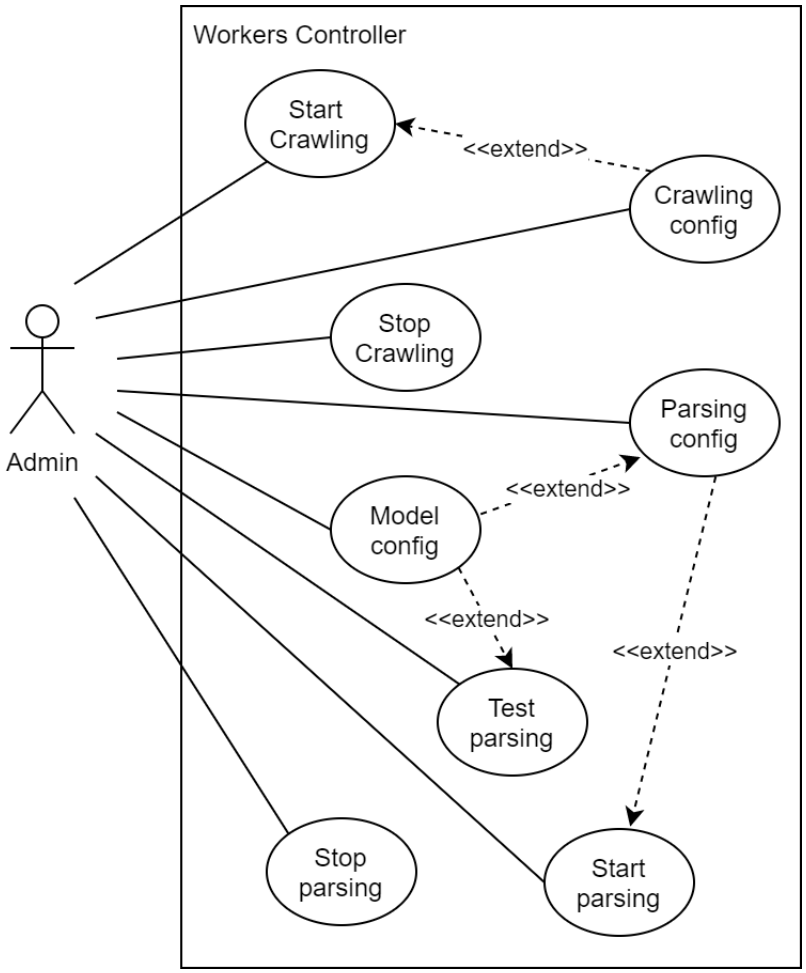


Figure 3.1: Use-case diagram for Web server

- **Use case specification: Crawling configuration**

Name	Crawling configuration
ID	CR01
Description	Before starting to crawl data, admin firstly setup parameters such as: date range of post, type, website, and limited number of post.
Pre-condition	N/A
Basic Flow	1. Select website. 2. Enable/disable anti-crawling resistance mode. 3. Enable post-date range. 4. Select date range of post. 5. Select type of post. 6. Select limit number of post.
Alternative Flow	At step 3, if admin disable date range, then ignore step 4
Exception Flow	N/A
Post-Condition	N/A

Table 3.1: Use case specification: Crawling configuration

- **Use case specification: Start crawling**

Name	Start crawling
ID	CR02
Description	After having done setup for crawling, admin can activate crawling process.
Pre-condition	User has done setup configuration for crawling.
Basic Flow	Click on Start Crawling button.
Alternative Flow	N/A
Exception Flow	N/A
Post-Condition	N/A

Table 3.2: Use case specification: Start Crawling

- **Use case specification: Stop crawling**

Name	Stop crawling
ID	CR03
Description	When workers are crawling, admin can stop all or each worker(s).
Pre-condition	There are at least 1 worker is in progress of task..
Basic Flow	Click on Stop Crawling button or Click on Stop button of each running workers to stop itself only.
Alternative Flow	N/A
Exception Flow	N/A
Post-Condition	N/A

Table 3.3: Use case specification: Stop Crawling

- **Use case specification: Parsing configuration**

Name	Parsing configuration
ID	PA01
Description	Before starting to parse data, admin firstly setup parameters such as: date range of post, type, website, and limited number of post.
Pre-condition	Have been done crawling so that database contains HTML data, and there is at least 1 worker free
Basic Flow	1. Select website. 2. Enable post-date range. 3. Select post-date range. 4. Enable crawl-date range. 5. Select crawl-date range. 6. Select type of post. 7. Select model. 8. Select record status. 8. Select limit number of post.
Alternative Flow	– At step 2, if user disable post-date range, then ignore step 3. – At step 4, if user disable crawl-date range, then ignore step 5.
Exception Flow	N/A
Post-Condition	N/A

Table 3.4: Use case specification: Parsing configuration

- **Use case specification: Start parsing**

Name	Start parsing
ID	PA02
Description	After having done setup for parsing, admin can activate parsing process.
Pre-condition	Admin has done setup configuration for parsing.
Basic Flow	Click on Start Parsing button.
Alternative Flow	N/A
Exception Flow	N/A
Post-Condition	N/A

Table 3.5: Use case specification: Start parsing

- **Use case specification: Stop parsing**

Name	Stop parsing
ID	PA03
Description	When workers are parsing, admin can stop all or each worker(s).
Pre-condition	There are at least 1 worker is in progress of task.
Basic Flow	Click on Stop Parsing button or Click on Stop button of each running workers to stop itself only.
Alternative Flow	N/A
68 **Exception Flow**	N/A
Post-Condition	N/A

Table 3.6: Use case specification: Stop parsing

- **Use case specification: Parser Model Configuration**

Name	Parser Model Configuration
ID	PA04
Description	There is a kind of custom parser model, that admin can modify to increase efficiency of parsing.
Pre-condition	N/A
Basic Flow	1. Choose a Parser Model that admin wants to modify. 2. Click on Add New Attributes. 3. Fill the necessary information of the model's attributes. 4. Click save.
Alternative Flow	N/A
68 **Exception Flow**	N/A
Post-Condition	N/A

Table 3.7: Use case specification: Parser Model Configuration

- **Use case specification: Parser Model Testing**

Name	Parser Model Configuration
ID	PA04
Description	Saved Parser Models need to be test carefully before being put into the system for data parsing processes. In here, admin can to that.
Pre-condition	There is at least one parser model saved in the database for testing.
Basic Flow	1. Admin search for HTML data records. 2. Select record(s) that admin want to test the parser model on. 3. Choose parser model. 4. Click Test Parsing.
Alternative Flow	N/A
68 **Exception Flow**	N/A
Post-Condition	N/A

Table 3.8: Use case specification: Parser Model Testing

3.4.1.b Normal User (User)

- Description: Normal users are those who want to sell their real estate on C2C e-commerce websites, and need to use a decision support system to choose the right selling price, or view relevant analysis. products for sale.
- Activities:
  - Enter a description of the real estate
  - View recommended real estate prices
  - Search real estate posts on many websites
  - View similar property
  - View statistics on the price fluctuations of real estate

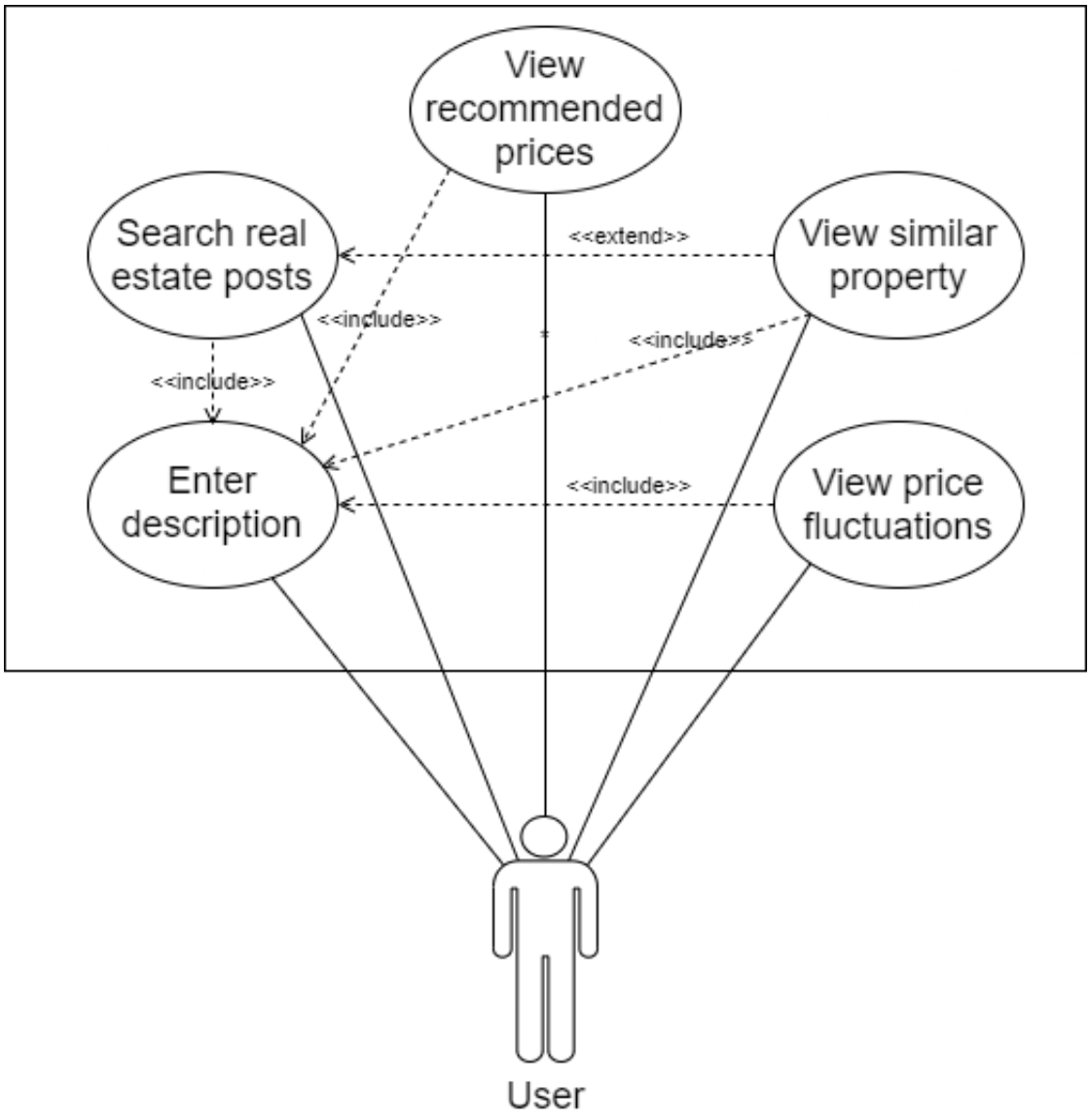


Figure 3.2: Use-case diagram for decision support system

- **Use case specification: Enter description**

Name	Enter description
ID	ED1
Description	User enters the description of the property to sell or buy.
Pre-condition	N/A
Basic Flow	Enter the description of the product want to sell or buy.
Alternative Flow	N/A
Exception Flow	User inputs wrong format or not input information. Ask user to re-enter description.
Post-Condition	The input process is completed and navigate to other task.

Table 3.9: Use case specification: Enter description

- **Use case specification: View recommended prices**

Name	View recommended prices
ID	SRP1
Description	User views suggested selling price based on entered description.
Pre-condition	The description of the property to be guessed at the price has been entered.
Basic Flow	View property price recommendations based on description entered.
Alternative Flow	N/A
Exception Flow	N/A
Post-Condition	N/A

Table 3.10: Use case specification: View recommended prices

- **Use case specification: Search real estate posts**

Name	Search real estate posts
ID	SP1
Description	Users search for posts of real estate for sale on e-commerce sites.
Pre-condition	The description of the real estate to be searched at has been entered.
Basic Flow	View posts of real estate currently for sale on many e-commerce sites based on descriptions entered.
Alternative Flow	View similar properties of the searched property.
Exception Flow	N/A
Post-Condition	N/A

Table 3.11: Use case specification: Search real estate posts

- **Use case specification: View similar property**

Name	View similar property
ID	SSP1
Description	Users view similar properties based on the description entered.
Pre-condition	The description of the real estate to find has been entered.
Basic Flow	View similar real estate for sale on many e-commerce sites nearby based on description inputted.
Alternative Flow	N/A
Exception Flow	N/A
Post-Condition	N/A

Table 3.12: Use case specification: View similar property

- **Use case specification: View price fluctuations**

Name	View price fluctuations
ID	VPF1
Description	Users view month-to-month price movements based on the description entered.
Pre-condition	The information of category of property and location has been entered.
Basic Flow	View price fluctuations through months.
Alternative Flow	N/A
Exception Flow	N/A
Post-Condition	N/A

Table 3.13: Use case specification: View price fluctuations

3.4.2 Activity Diagram

3.4.2.a Administrator of Workers system (Admin)

- **Crawling data activity**

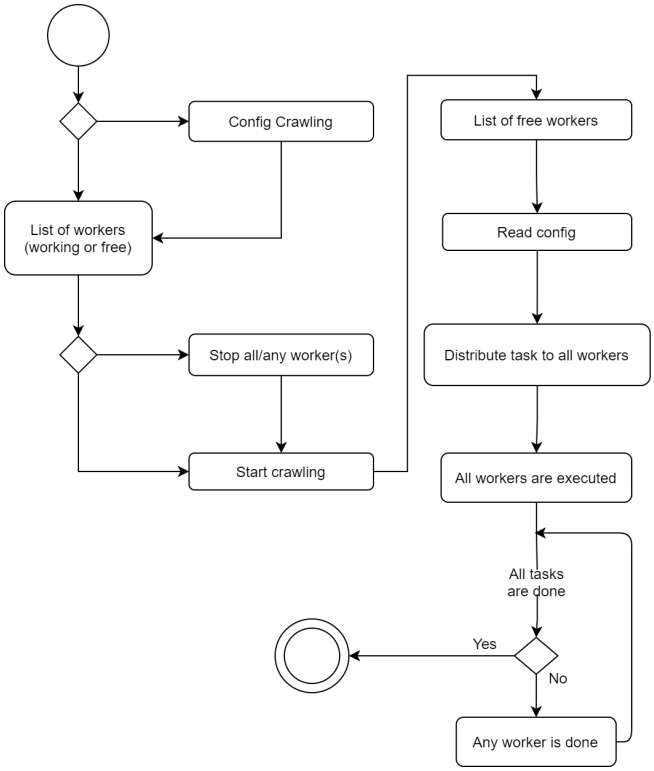


Figure 3.3: Activity diagram of crawling data.

Description

Step 1. Firstly, admin configures parameters for crawling, this step is optional. Without this step, crawling process will be started using default setting.

Step 2. There may be some workers are in progress, or even all are busy. Hence, admin are able to stop any running worker, or the whole.

Step 3. Admin clicks on Start Crawling to initiate the process.

Step 4. After that, the system will list free worker(s) which is/are ready to receive task.

Step 5. Then, the system reads configuration setting to distribute tasks to workers in the list.

Step 6. When all tasks are delivered, system is in the state of listening completion response of workers. When all are completed, system will notify and show the report of the whole process to admin.

- **Parsing data activity**

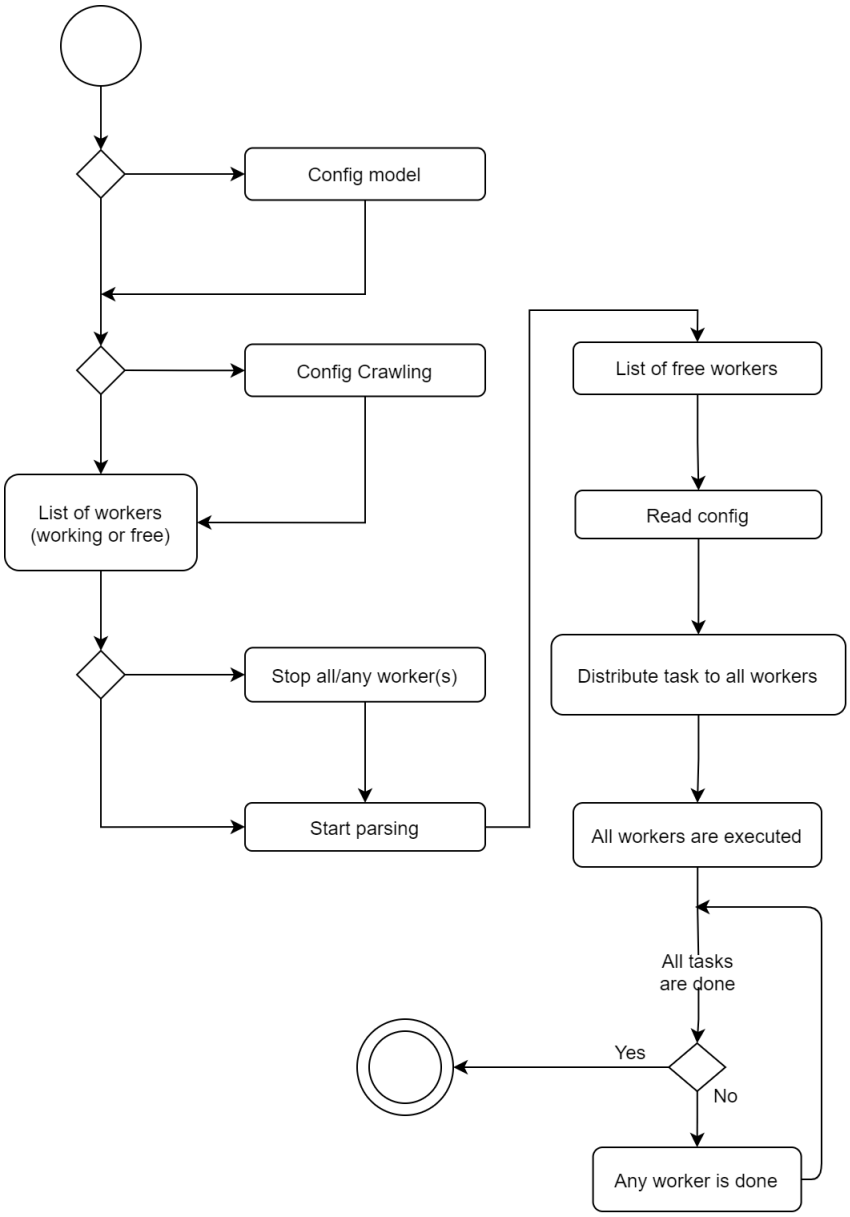


Figure 3.4: Activity diagram of parsing data.

Description

Step 1. Firstly, admin configures Parser Model, this step is optional. Passing this step, parsing process will be started using already saved models.

Step 2. Admin configures parameters for parsing, this step is also optional. Without this step, crawling process will be started using default setting.

Step 3. There may be some workers are in progress, or even all are busy. Hence, admin is able to stop any running worker, or the whole system.

Step 4. Admin clicks on Start Parsing to initiate the process.

Step 5. After that, the system will list free worker(s) which is/are ready to receive task.

Step 6. Then, the system read configuration setting to distribute tasks to workers in the list.

Step 7. When all tasks are delivered, system is in the state of listening completion response of workers. When all are completed, system will notify and show the report of the whole process to admin.

- **Testing parser model activity**

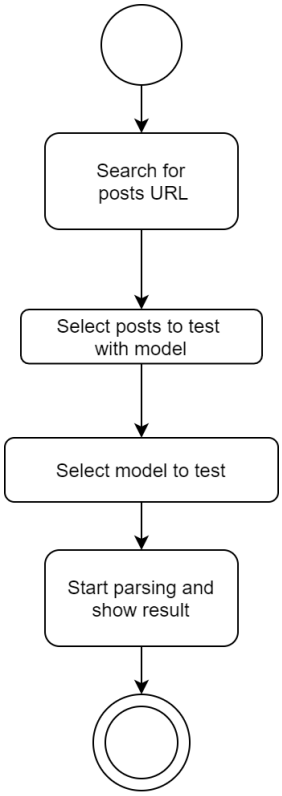


Figure 3.5: Activity diagram of testing parser model.

Description

Step 0. Before testing parser model, there must be at least one model saved and HTML post data in the database.

Step 1. Admin select value of filters of search tool, then click Search to query HTML post record from database.

Step 2. After a period of loading time, there will be list of records for admin to select.

Step 3. Then, admin choose model to be tested and click on Test button.

Step 4. Finally, the result of parsing data will be shown.

3.4.2.b Normal User (User)

- **Activity diagram for recommendation system**

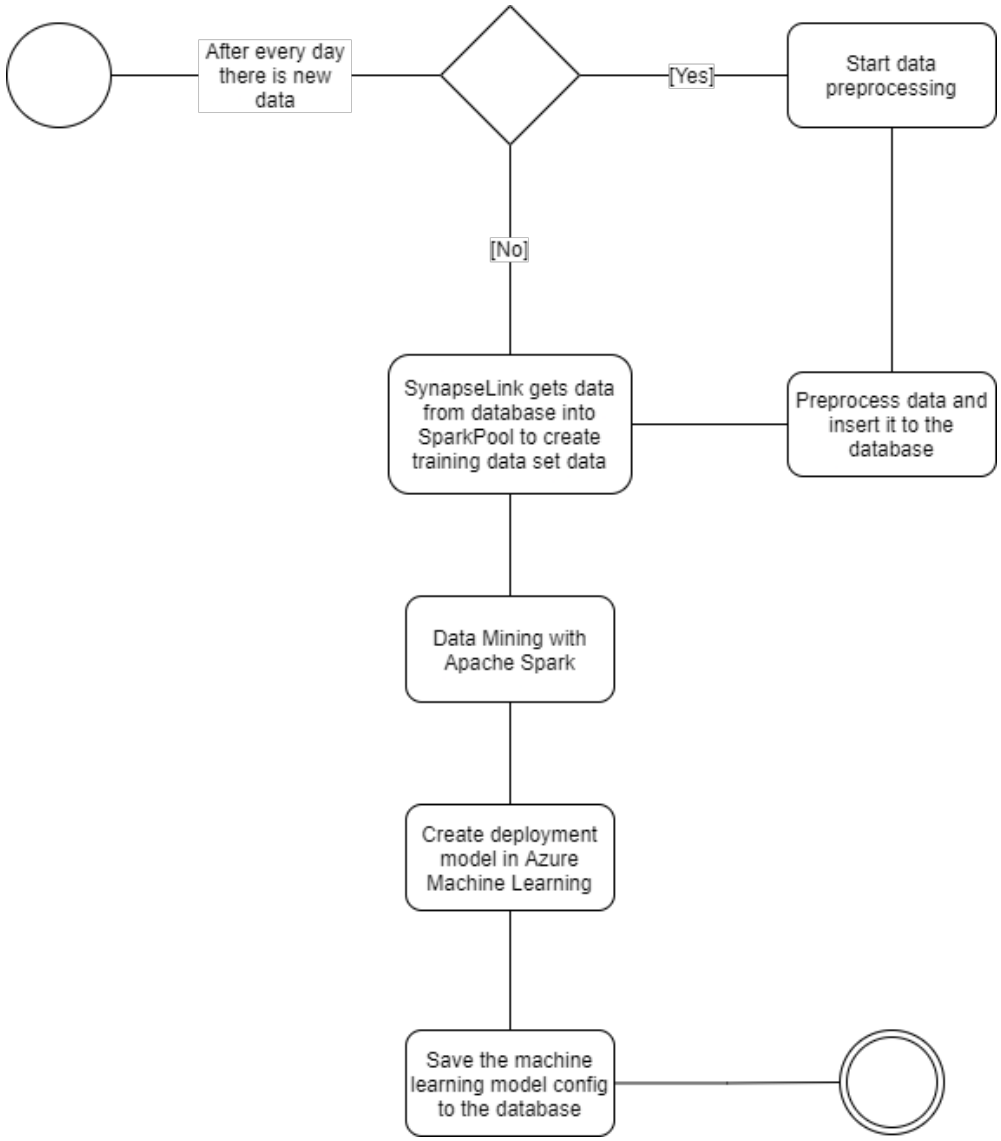


Figure 3.6: Activity diagram for recommendation system

Description

Step 1: Start data preprocessing process.

Step 2: The parsed data will be preprocessed to normalize the data on different pages and then inserted into database.

Step 3: Once normalized, that data will be pushed into SparkPool by SynapseLink from the database.

Step 4: Apache Spark is responsible for data mining and Azure Machine Learning for creating deployment machine learning model on saved data.

Step 5: After the model has been generated for the data set, the model will be saved to the database and used to predict for following queries.

- **Activity diagram for view recommended prices**

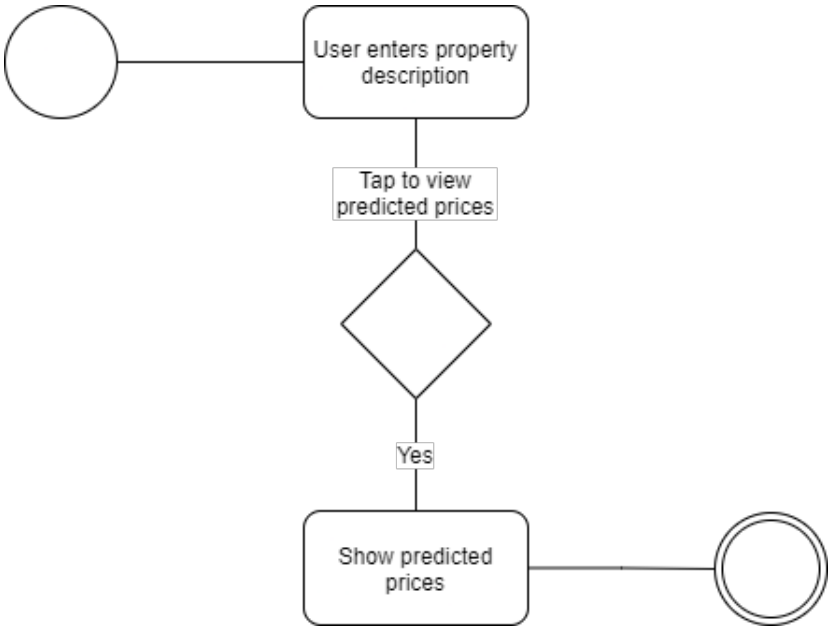


Figure 3.7: Activity diagram for viewing recommended prices

Description

Step 1: User enters description of the property to be sold.

Step 2: User tap to view predicted prices.

- **Activity diagram for search real estate posts**

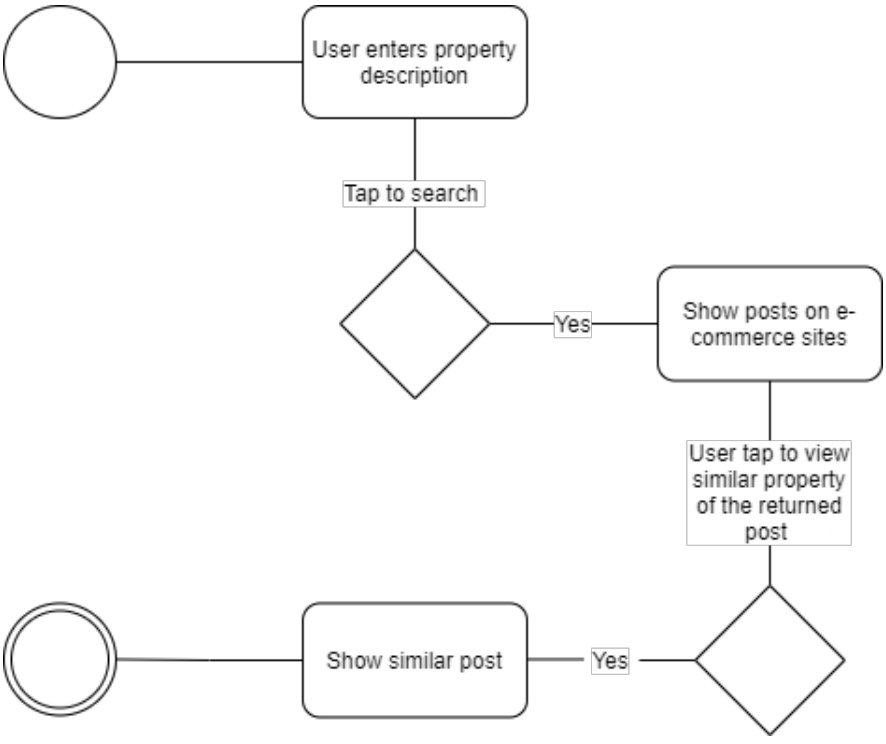


Figure 3.8: Activity diagram for searching real estate posts

Description

Step 1: User enters description of the property to be sold.

Step 2: User tap to search.

Step 3: When the results appear, the user can view more similar property of the returned post.

- **Activity diagram for view similar property**

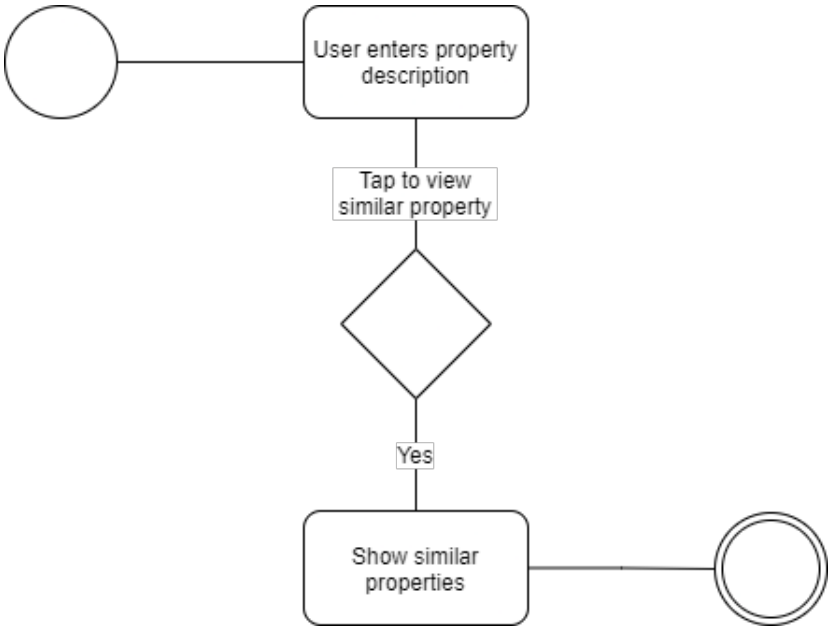


Figure 3.9: Activity diagram for viewing similar property

Description

Step 1: User enters description of the property to be sold.

Step 2: User tap to view similar properties currently for sale on e-commerce sites.

- **Activity diagram for viewing price fluctuations**

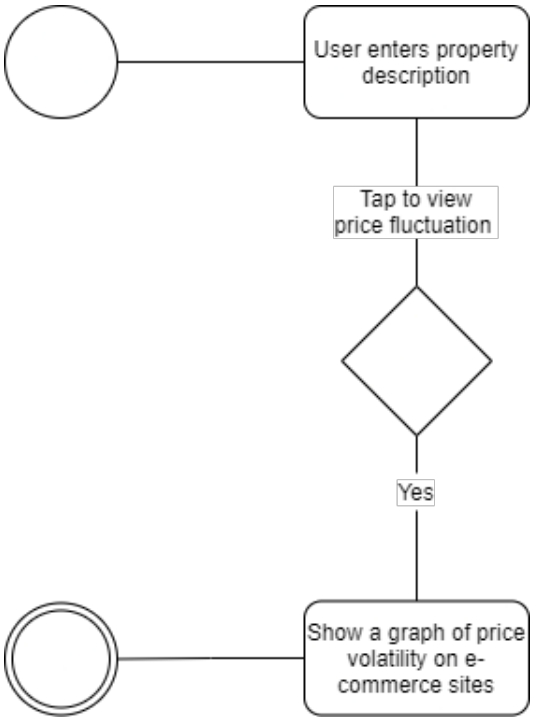


Figure 3.10: Activity diagram for view price fluctuations

Description

Step 1: User enters description of the property to be sold.

Step 2: User tap to view view price fluctuations of property on e-commerce sites.

4 System Design and Analysis

4.1 General architecture

a. How the system works:

E-commerce websites (chotot.vn, batdongsan.com.vn, nhadat247.com.vn...) have thousand of posts per day. These posts will be collected by various distributed data collection components and sent back to the data warehouse. Here, data will be extracted and processed to clean data for the mining process. The system uses large data storage technologies in response to the fast data generation rate.

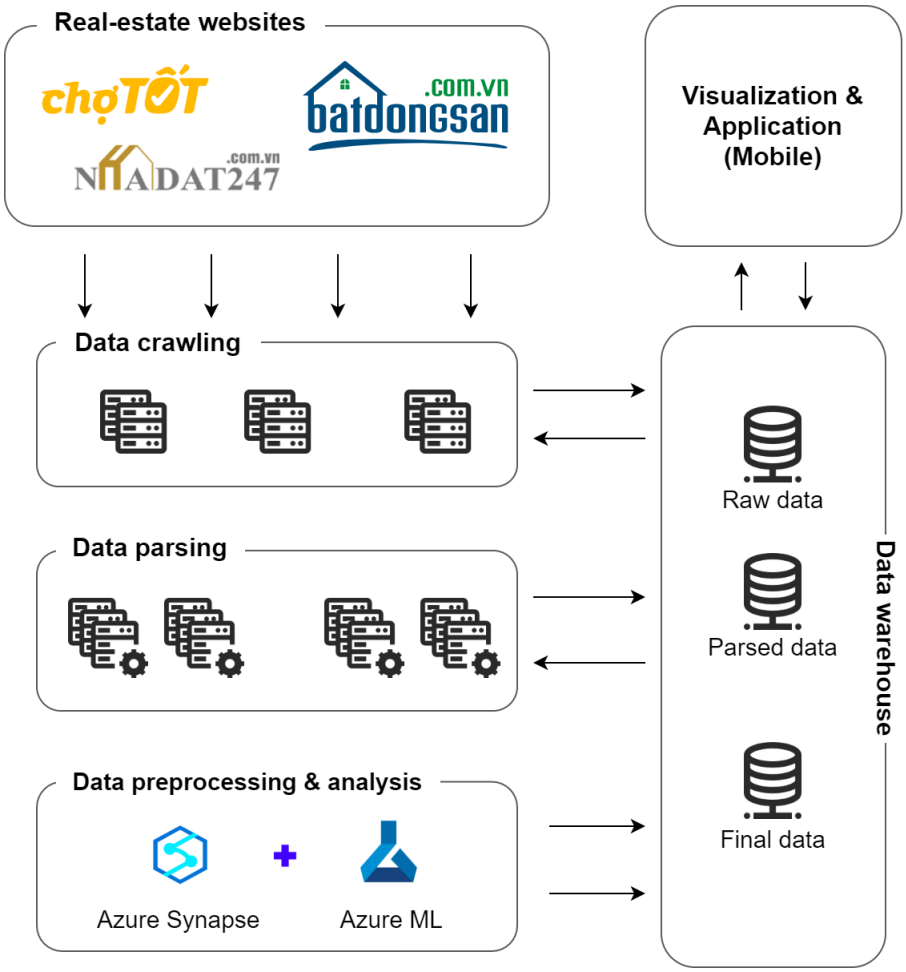


Figure 4.1: General Architecture.

b. The system's architecture consists of 4 main components:

- **System of workers:**

- The component contains two parts, Data Crawling, and Data Parsing. This is a system of machines to collect HTML source code and extract data. The HTML collection consists of several workers that can operate decentralized across multiple locations and run on multiple platforms. HTML source code is scraped from real-estate websites using a crawler algorithm then stored in the data warehouse. The Data Parsing service is responsible for extract HTML source code to data that contains attributes of a property. It obtains HTML stored from the data warehouse then extracts and sends parsed data back to the data warehouse component.

- **Data preprocessing and analysis:**

  - The component consists of the process of normalizing data and analyzing as well as applying machine learning to data. The parsed data from the data warehouse is just basically cleaned, so it needs to be normalized so that the computer can use the resource. The normalizing step occurs on Azure Synapse Analytics, and its output is the final data for the machine learning which is powered by Azure Machine Learning.

- **Data warehouse:**

  - Data warehouse component is responsible for storing data for the whole system. The storage component is powered by Azure Cosmos DB which has the advantages such as: the ability to store data on a global scale, guaranteed low latency, Always available and flexible scalability.

- **Visualization and Application:**

  - A place to provide interfaces and applications for users. Applications using the collected and analyzed data will be visually displayed to help users make easy decisions

4.2 Data crawling component

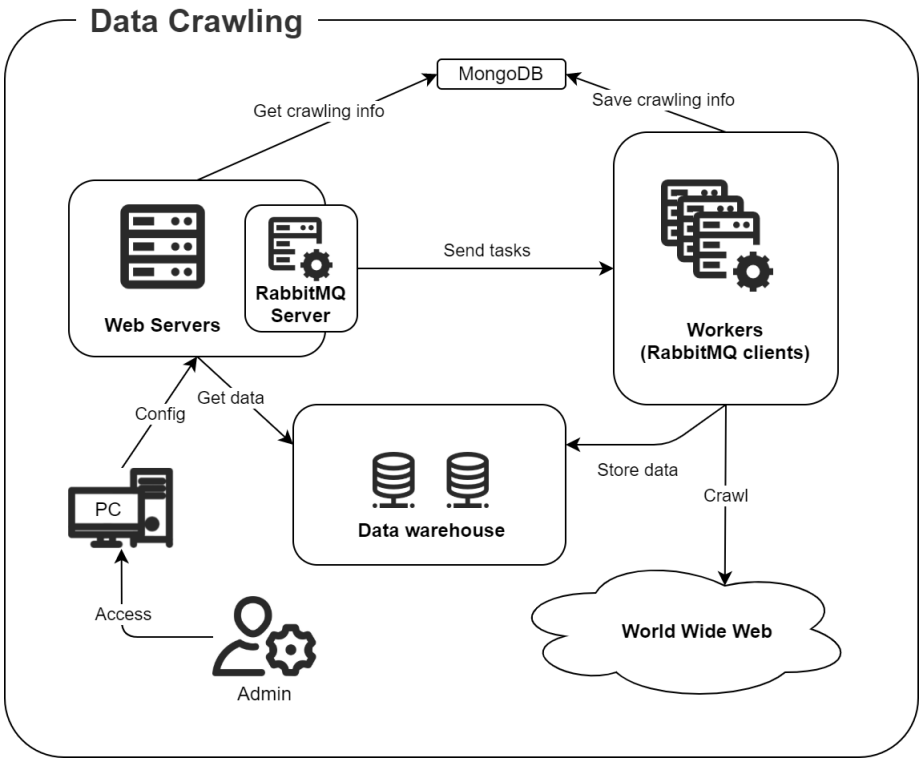


Figure 4.2: Data Crawling component

The figure above describes the architecture of data crawling component which includes:

- **Web server:**

  - This part is the site where administrators of the system are able to control every things. Beside the function of provide UI for admin to easily control system of distributed workers, it is responsible for receive configuration from admin to deliver task to workers using RabbitQM. Moreover, data that is necessary for worker to do task, will be sent to a database agent (powered by MongoDB) so that workers can access and obtain needed data.

- **Distributed workers**:

  - They are RabbitMQ-client computers that are responsible for crawling data according to tasks delivered by the server. The system of distributed crawlers helps to avoid being recognized as a crawler bot from the target server which is a real-estate website. When crawling HTML data, workers continuously save data to the data warehouse. Moreover, the information of task is also sent to the MongoDB

database agent for server to get and shown to admin.

- **Data warehouse**:

  - Data watehouse is the place of storing data for the whole system, which is powered by Azure Cosmos DB. It provide APIs that other component can use to do query on the database.

The data collection process will be performed simultaneously by many different machines. Therefore, it is necessary to have a system of distributed crawlers and to manage the work performed by each worker to ensure that no process is overloaded or idle. In the current system, we set up the delivery and job management mechanism using RabbitMQ. In particular, RabbitMQ Server is responsible for dividing work for RabbitMQ Clients, which are processes on many different machines, to perform HTML collection. An overview of the data crawling process can be presented as follows: The administrator can configure the crawling parameters such as date range, type of post, the number of URLs to crawl, etc., and finally start the crawling process. The Web Server will then use the configuration parameters to start and control the RabbitMQ Server to coordinate work for the RabbitMQ Clients. After the data is collected, it will be sent to the data warehouse for further processing.

On commercial real estate websites, there are two types of pages to be considered: post pages and index pages. The post pages are places where the information of properties is display on the web for users to read, they are also what we want to crawl. The index page is a page that contains a list of posts. On this page, we can get URLs of others index pages and URLs of post pages. Therefore, if we process the homepage of a website we can reach all posts of the website to crawl. The idea of crawling is described as figure below:

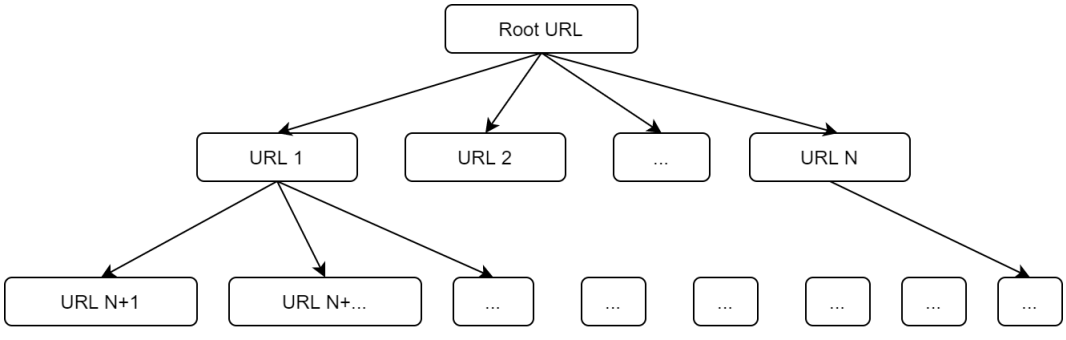


Figure 4.3: URL tree graph

The Root URL is the homepage of a website or a category page that contains

all URL post of a type, for example: https://batdongsan.com.vn/ban-can-ho-chung-cu. From root URL, we can request and extract to get its children URLs, then we repeat the process on each of the children URLs to get the next level of URLs to the end of the website.

We have studied about the diversity of real-estate websites in Vietnam. Then, we see that there are three type of websites which have different angles, so that we proposed solution for each type:

Type 1: Websites using the HTTP GET method will get the HTML source code containing all the information displayed on the web. This type is the easiest one to crawl, we just sent request and get HTML data.

Type 2: Websites using the HTTP GET method will get the HTML source code containing only the JavaScript code. These snippets only generate HTML code that contains the information displayed on the web when loaded by the browser. In this case, we use Selenium which is a library help control browser like Chrome or Firefox to render the web page.

Type 3: Websites using the HTTP GET method will get the HTML source code containing some information and the rest will be loaded with JavaScript when user interact with the website. With this case, Selenium is also our solution, it helps humanly interact with the website such as scrolling up/down, click, etc.

Although websites are different about crawling method, our goal is develop a machanism that works with all type of website mentioned. The flow of crawling is described as figure below:

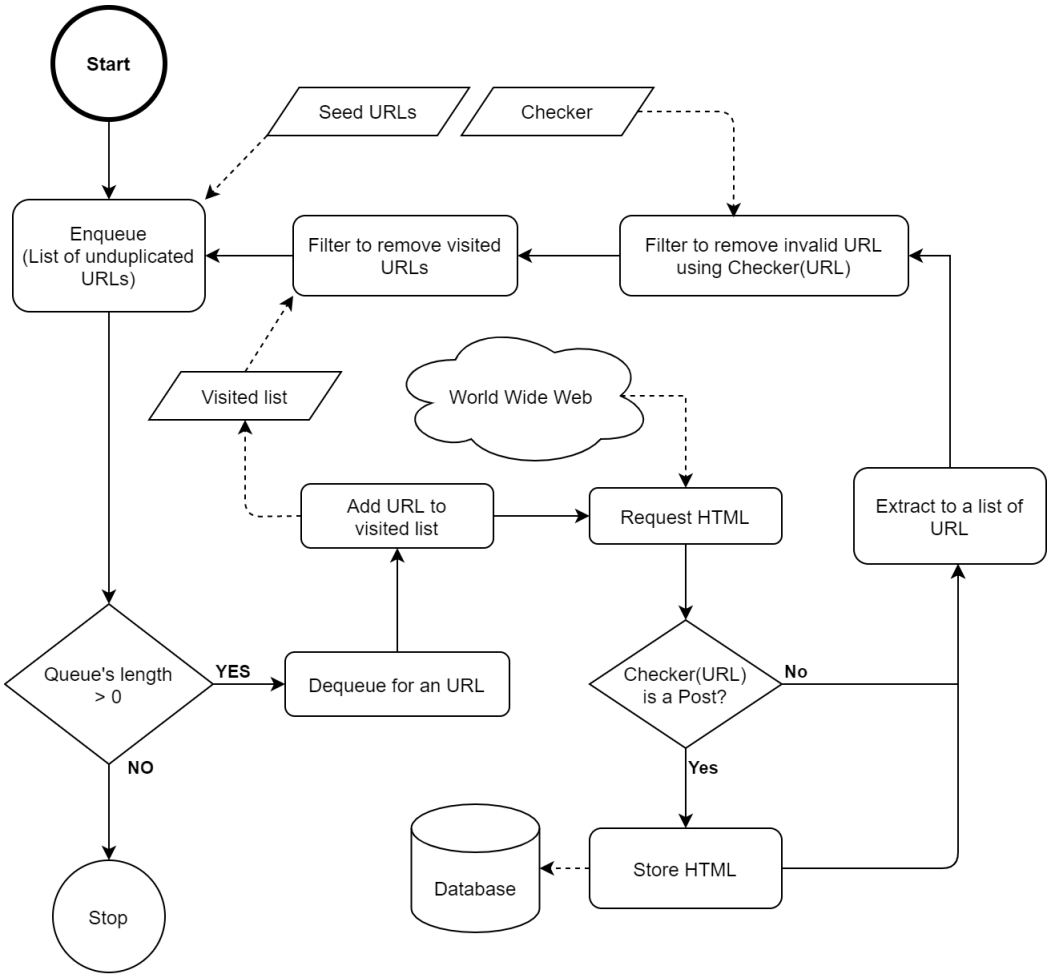


Figure 4.4: Flowchart of Crawling component

Step 1: Starts with a given Seed URLs and a Checker. The system will put the seeds URL into the queue. Note that the queue contains unduplicated URLs.

Step 2: The system goes into a loop that will end if length of the queue is zero.

Step 3. When inside the loop, the system start with get an URL, called current URL, from the queue by dequeue method. The current URL, which is used to request HTML page source, is added to a list called Visited list.

Step 4: The system use the Checker object to check whether the URL is a post page or an index page. If it is a post page, then it will be store in the data warehouse. Afterward, move on to the next step.

Step 5: At this step we extract URLs in the obtained HTML page source.

Step 6: We reuse the Checker object to check on each URL whether it is valid or not. We remove URLs which are invalid. With each of valid URLs, we continue to check whether it is in the Visited list or not. If any URL is visited, we remove it. Finally we get the Result list of URL.

Step 7: To put each URL of the Result list into the queue, we have to make sure that it has not been included in the queue before. Then the system finish the current loop and move to the next loop.

4.3 Data parsing component

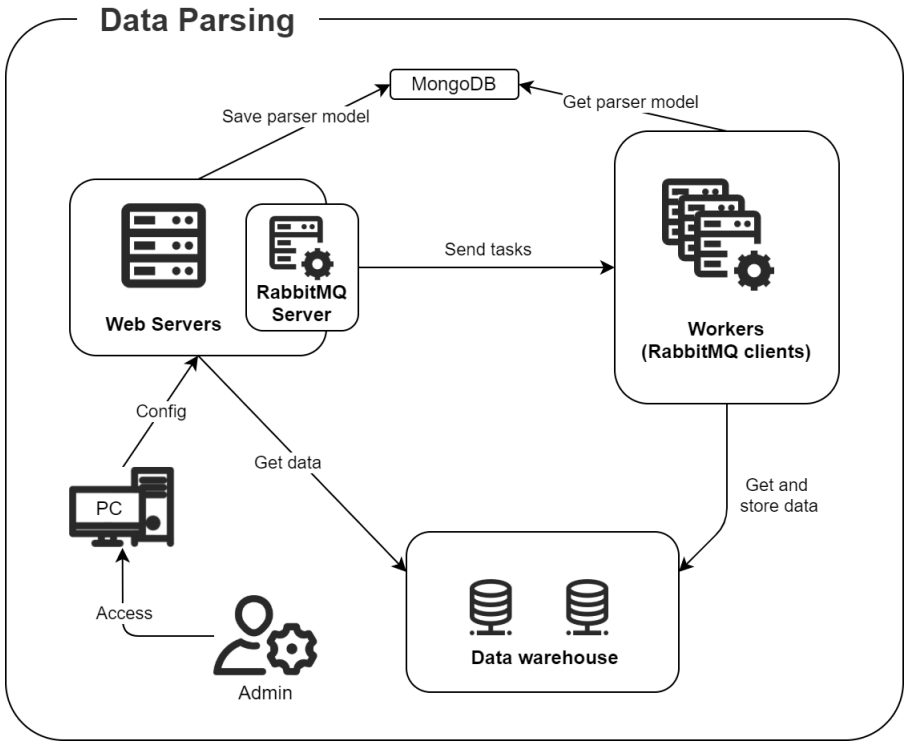


Figure 4.5: Data Parsing component

The figure above describes the architecture of data parsing component which includes:

- **Web server:**

  - This part is the site where administrators of the system are able to control every things. Beside the function of provide UI for admin to easily control system of distributed workers, it is responsible for receive configuration from admin to deliver task to workers using RabbitQM. Moreover, web server is a place where admin configure Parser Model and save them into to a database agent (powered by MongoDB) so that workers can access and obtain when doing data crawling.

- **Distributed workers**:

  - They are RabbitMQ-client computers that are responsible for parsing data accord-

ing to tasks delivered by the server. When starting, workers load Parser model from the database agent. Then, when doing parse HTML data, workers continuously save data to the data warehouse. Moreover, the information of task is also sent to the MongoDB database agent for server to get and shown to admin.

- **Data warehouse**:

  - Data watehouse is the place of storing parsed data.

When initiating the extraction of the collected HTML data, the Web Server checks to see if there is any unextracted data in the database. If there is a Web Server transmits those data to the worker responsible for this process to perform the extraction. This worker will read the configuration data from the database to perform data extraction and store the extracted data in the database.

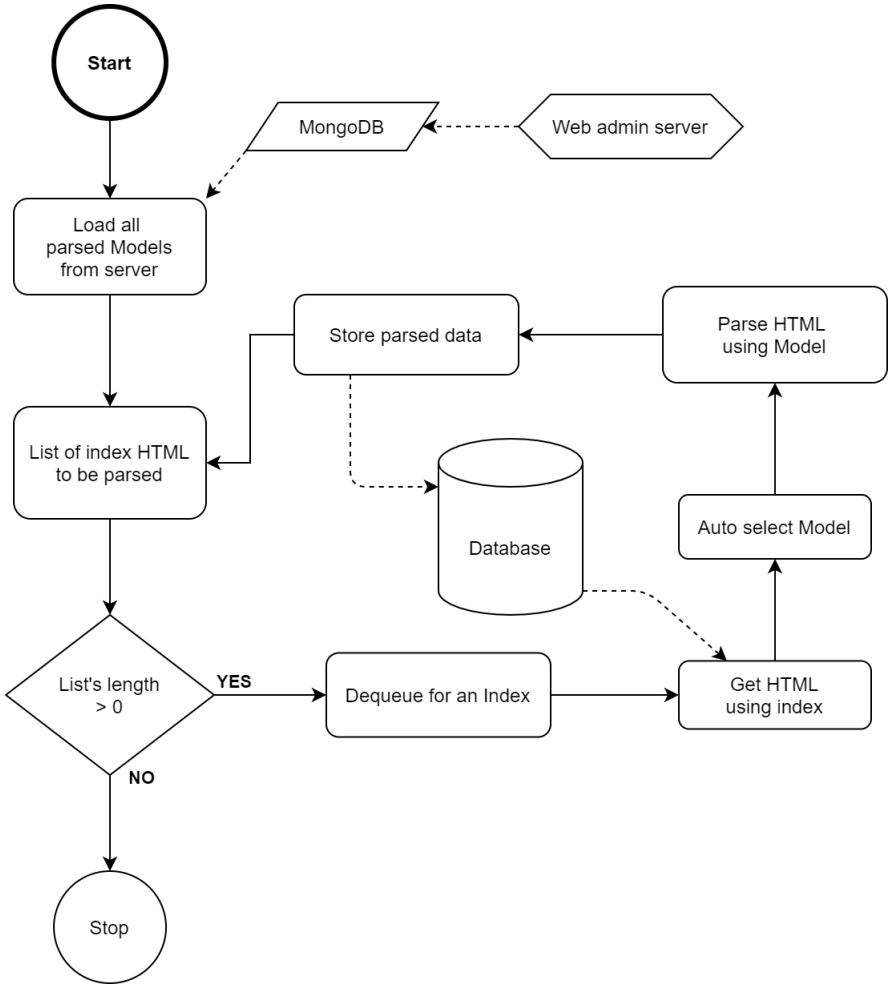


Figure 4.6: Flowchart of Data Parsing component

In developing the data parsing component of the system, the data after being collected will be in the form of raw HTML text, on which the computer cannot perform analysis or statistics. In other words, the raw HTML text data does not bring any value

to the system in terms of information until it is extracted into pre-defined attributes so that the system can easily manipulate. Hence, this information becomes useful. Therefore, it is necessary to develop a model to extract information from the collected data. Basically, extracting information in an HTML document is quite simple because HTML documents are organized according to the structure and tag system, particularly in the HTML language. If we know the exact location of the content to be extracted, we can completely get that content. So, extracting HTML content is simply giving definitions of paths that correspond to the contents of the attributes to be retrieved so that the parser component can use them when receiving parsing tasks.

The problem here is that each different website will have a different interface, even a website can have many interfaces for different types of products. The changing interface means that the structure and arrangement of HTML text tags also change. Therefore, we propose two methods:

- **Method 1:** For each type of interface, we need to define a parser corresponding to it. In addition, this work, in addition to the step of getting from the path, sometimes requires many other complicated processing steps to get the desired content. The model attempts to create a generic definition for all possible scenarios encountered when attempting to extract content. The advantage of the method is that it can parse exactly the information that we need, the efficiency is about 99%. However, this method does not solve the problem if the admin of the website dramatically change a part or event the whole website, so that our Parser model will be failed when parsing data cause the efficiency will be low or even zero.
- **Method 2:** We see that, content of an property (price, size, saler, address, etc) on real-estate websites, is shown in a structured way. Every type of attributes has an defined structure. Therefore, Natural Language Processing is another better solution and SpaCy is the technology that we has applied. Spacy provides a method to process an text using multiple of pipelines. One of them is NER, Named Enity Recognition, which is defined by a rule of grammar. As we assume that each attributes of a property in its post page is an Entity. Hence, we can use the utility of SpaCy to train NLP model that can parse text content of property post page into Entites as well as attributes. Beside its advantages, this method has a drawback that its accuracy is not actually high, because the input text can contain mistakes in spelling, typing errors or missing of rules.

4.4 Data pre-processing component

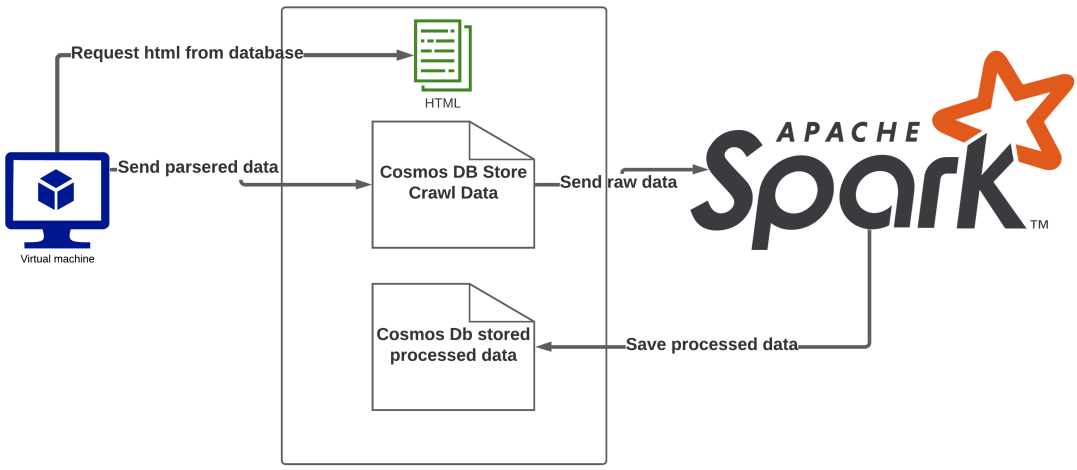


Figure 4.7: Architecture of preprocessing component.

Figure above shows the data preprocessor component architecture, including:

- **WebServer**: perform the process of extracting and coordinating processed data into each reasonable data container.
- **Cosmos DB Store Crawl Data**: is a component that contains all the items that is parsed by webserver.These items is divided into 2 containers which are update_data and parser_data by specific conditions.
- **Cosmos Db store processed data**: is a container that contains all the items is cleansed and normalized by Apache Spark.
- **Apache Spark** component used to preprocess data, clean raw data from *Cosmos DB Store Crawl Data* component and then clean data into *Cosmos Db store processed data* component.

In general, the Web Server will take the data that has been extracted but not pre-processed and transmit it to the Preprocessing Worker cleansing the data. That raw data will be classified into 2 containers: data that exist in final container which need to be updated and new data. When starting the data mining process, the system uses Spark/Mlib to perform mining on Azure Synapse which generates information and stores this information in the database.

At this step, Azure Synapse Link for Azure Cosmos DB, which is a cloud-native hybrid transactional and analytical processing (HTAP) capability that enables you to run

near real-time analytics over operational data in Azure Cosmos DB, is used to create a tight seamless integration between Azure Cosmos DB and Azure Synapse Analytics.

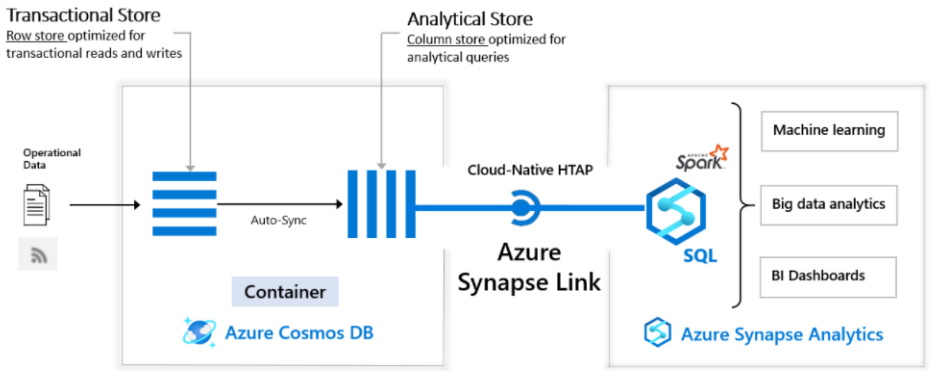


Figure 4.8: Architecture of Azure Synapse Link.

To analyze large operational datasets while minimizing the impact on the performance of mission-critical transactional workloads, traditionally, the operational data in Azure Cosmos DB is extracted and processed by Extract-Transform-Load (ETL) pipelines. ETL pipelines require many layers of data movement resulting in much operational complexity, and performance impact on your transactional workloads. It also increases the latency to analyze the operational data from the time of origin.

When compared to the traditional ETL-based solutions, Azure Synapse Link for Azure Cosmos DB offers several advantages such as:

- **Reduced complexity with No ETL jobs to manage**: Azure Synapse Link allows you to directly access Azure Cosmos DB analytical store using Azure Synapse Analytics without complex data movement. Any updates made to the operational data are visible in the analytical store in near real-time with no ETL or change feed jobs. You can run large scale analytics against analytical store, from Azure Synapse Analytics, without additional data transformation.

- **Near real-time insights into your operational data**: ETL-based systems tend to have higher latency for analyzing your operational data, due to many layers needed to extract, transform and load the operational data. With native integration of Azure Cosmos DB analytical store with Azure Synapse Analytics, you can analyze operational data in near real-time enabling new business scenarios.

- **No impact on operational workloads**: With Azure Synapse Link, you can run analytical queries against an Azure Cosmos DB analytical store (a separate column

store) while the transactional operations are processed using provisioned throughput for the transactional workload (a row-based transactional store). The analytical workload is served independent of the transactional workload traffic without consuming any of the throughput provisioned for your operational data.

- **Optimized for large-scale analytics workloads:** Azure Cosmos DB analytical store is optimized to provide scalability, elasticity, and performance for analytical workloads without any dependency on the compute run-times. The storage technology is self-managed to optimize your analytics workloads. With built-in support into Azure Synapse Analytics, accessing this storage layer provides simplicity and high performance.
- **Cost effective:** With Azure Synapse Link, you can get a cost-optimized, fully managed solution for operational analytics. It eliminates the extra layers of storage and compute required in traditional ETL pipelines for analyzing operational data.

The data collected and extracted is only very rudimentary data, if directly storing this data will be very difficult in the data mining process or even make the mining process inaccurate. Therefore, we need a mechanism to preprocess the data, which is data from e-commerce websites including chotot.vn, nhadat247.com.vn and batdongsan.com.vn. There are many data problems that need to be preprocessed such as:

- The price of the property has many different formats such as: 1.000.000.000 VND, 1 Ty, 900 Trieu...
- The format of address and category in each e-commerce websites is different.
- The posts are missing a lot of important data: phone number, price, address, surface, district ...

To solve the above data problems, we have done the following:

- For property prices presented in different formats, we have developed a syntax set of price formats from websites, so that price formats will be standardized by this toolkit. 1 unique form is a string of float number(1.9) with unit is 1 billion VND.
- We have developed the code to translate from many different formats of each website into a fixed form, then filter from the address to elements such as ward,... and divide the category of the property into fixed forms such as house, flat,....
- For posts that are missing a lot of important data, the team is choosing to ignore these posts and treat them as noisy data.

4.5 Architecture for storing and processing big data of the system

In Vietnam, the number of e-commerce websites is very large, the capacity is up to terabytes of data, and that data is updated and changed continuously by millions of users every day, every hour, every job How to store and analyze data efficiently, quickly, and minimize costs is a big challenge. For such large data, when applying the traditional database system, the following problems will arise: When the amount of access to the database is too large, the database will be overloaded with reading and writing, we have to create a queue to batch process each batch of hits to reduce the load on the database for a unit of time. That creates the additional problem that as traffic increases, we need more queues, leading to bottlenecks when there is only one database.In the traditional solution, we have to multiply many servers, each server holds a piece of data, this creates the difficult task of managing synchronization, directing the amount of access between servers and increasing the time. data access time due to having to handle the above tasks. In addition, the difficulty of the above work also leads to the error of the administrator, leading to asynchronous data and an unbalanced system load between servers. In addition, when there are many servers, the probability of hardware failure will be higher, we have to manage backups.

Today's applications are required to be highly responsive and always online. To achieve low latency and high availability, instances of these applications need to be deployed in datacenters that are close to their users. Applications need to respond in real time to large changes in usage at peak hours, store ever increasing volumes of data, and make this data available to users in milliseconds.

Azure Cosmos DB is Microsoft's fast NoSQL database with open APIs for any scale. The service is designed to allow customers to elastically (and independently) scale throughput and storage across any number of geographical regions. Azure Cosmos DB is the first globally distributed database service in the market today to offer comprehensive service level agreements encompassing throughput, latency, availability, and consistency. Catalog data usage scenarios involve storing and querying a set of attributes for entities such as property. Attributes for this data may vary and can change over time to fit application requirements.Azure Cosmos DB supports flexible schemas and hierarchical data, and thus it is well suited for storing product catalog data.

Data Modeling

First, there are many ways to describe Azure Cosmos DB, but for our purposes, we can define it as having two primary characteristics: it is horizontally scalable, and it is non-relational

- **Horizontally scalable**: Azure Cosmos DB, data is stored in a single logical container. But behind the container, Azure Cosmos DB manages a cluster of servers, and distributes the workload across multiple physical machines. This is transparent to us – we never worry about these back-end servers – we just work with the one container. Azure Cosmos DB, meanwhile, automatically maintains the cluster, and dynamically adds more and more servers as needed, to accommodate your growth. And this is the essence of horizontal scale.

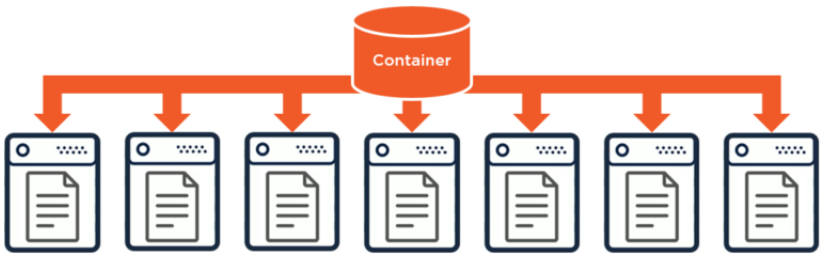


Figure 4.9: Data Modeling characteristic.

Each server has its own disk storage and CPU, just like any machine. And that means that – effectively – you get unlimited storage and unlimited throughput. There's no practical limit to the number of servers in the cluster, meaning no limit on disk space or processing power for your Cosmos DB container.

- **Non-relational**: data is stored in documents – that is JSON documents. Now there's certainly nothing can store in a row that you can't store in a JSON document.

"Speed and performance" is the name of the game in Azure Cosmos DB, with comprehensive SLAs on availability, throughput, latency, and consistency. Reads and writes have extremely low, single-digit millisecond latency – meaning that they complete within 9 milliseconds or less in most cases. So, in order to deliver on these performance guarantees, documents that be written to a container are very likely to be stored across multiple physical servers behind the scenes.

Architecture of warehouse

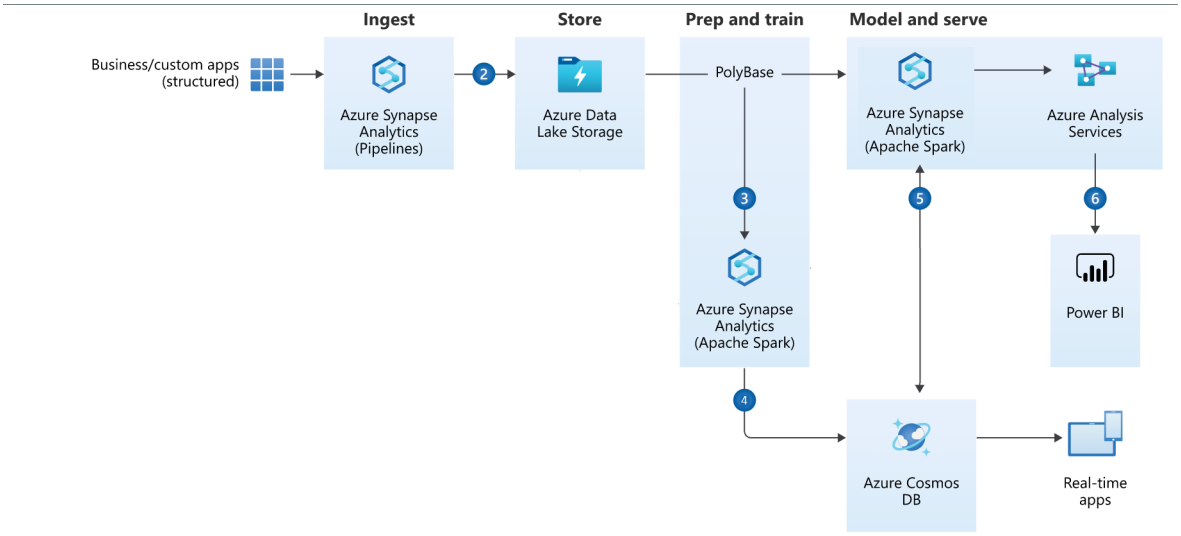


Figure 4.10: Data warehouse architecture.

Based on the characteristics of the system,Enterprise Data Warehouse Architecture from Microsoft is very suitable to build the system that bring together all real estate data at any scale easily, and to get insights through analytical dashboards, operational reports, or advanced analytics for all users.
The architecture contains the following components:

- **Data source**
  - **Azure CosmosDB**:Data after being parsed from html documents is saved to cosmosdb for data cleaning and normalization.
- **Ingestion and data storage**
  - **Synapse pipeline** allows to create, schedule and orchestrate your ETL/ELT workflows and normalization.It is in charge of transferring real estate data from the data source to the data analysis tools and is in charge of transferring again the processed data back to the database.
- **Analysis and reporting**
  - The duties of this component in the system include:
    - Cleansing and normalizing data.
    - Remove noisy data.
    - Prepare training dataset, training machine learning model and prepare data for visualization.

- **Apache Spark** includes many language features to support preparation and processing of large volumes of data so that it can be made more valuable and then consumed by other services within Azure Synapse Analytics.Spark pools in Azure Synapse Analytics also include Anaconda, a Python distribution with a variety of packages for data science including machine learning. When combined with built-in support for notebooks, you have an environment for creating machine learning applications.
- Power BI is a suite of business analytics tools to analyze data for business insights. In this architecture, it queries the semantic model stored in Analysis Services.

After the Data is parsed and saved to cosmosdb, the pipelines from Synapse that have been installed from scratch feed the raw data from the storage containers into processing at Azure Synapse. At this step, Azure Apache Spark will initially clean the data, return it to a normalized form first, and then remove the noisy data. Clean data is saved back into the container for user queries through pipelines.

For model training for machine learning, clean data from CosmosDb will be queried from Azure Apache Spark, creating a training data set without affecting the data in the container, avoiding user error. After that, Spark will train models to register those models to Azure Machine Learning for usage purposes.

4.6 Architecture for recommendation system

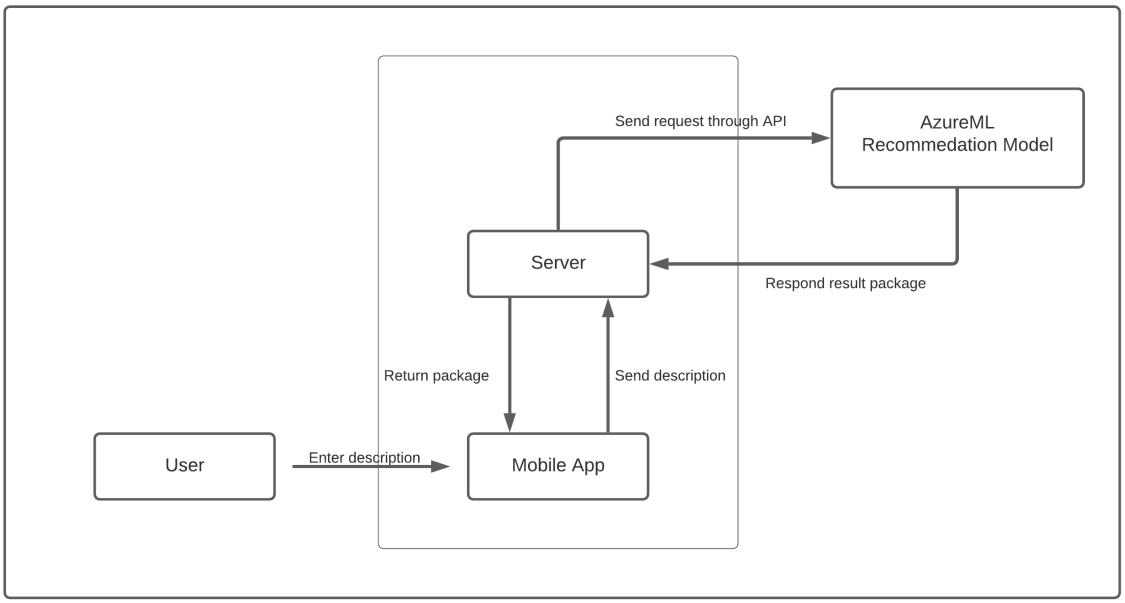


Figure 4.11: Architecture of recommendation system

Currently, businesses are paying more attention to big data, considering it as one of the factors that have an important influence on their business situation and development in the future. After the pioneering steps of science, not only scientists but business people have access to big data sources, and businesses have begun to be able to exploit it. Particularly in Vietnam, currently, the number of C2C websites in Vietnam is increasing very quickly and is diverse in categories. From the collected e-commerce big data, we can analyze and process many new problems to get useful information about the market on that data, such as predicting buying trends, sell, statistics, general business situation, commodity price fluctuations.

In this topic, a problem that needs to be solved is suggesting a price of property and provide some methods that support people on research and compare several properties. Because there are so many e-commerce websites with different characteristics, when a person wants to buy or sell their products, they have to spend a lot of time to find the right websites to make a decision.The need for a system to help users make quick decisions is essential.

To build such a system, based on the amount of data collected from many e-commerce real estate websites, the system will process and analyze to give useful information to users. The user must enter the necessary data to use the recommendation

system:

- Location, surface, width, length, number of bedrooms and toilets...
- For example: "Home, Ho Chi Minh, District 10, 100m^2, 10m, 10m...".

The main type function of recommendation system: Predict price. On the predict price function, there are 4 types of price which will be predicted: the general price is predicted on the data of the whole database, the price which is contains only data from chotot, nhadat247 or batdongsan.With 4 types of prices offered from the system, users will consult, compare and make their own decisions about the price of these property.

Property price recommendation system is built on linear regression model based on the input description of user.However, there are some attributes that are not suitable for input type of linear regression model such as: district, region, category..which are in String type. Therefore, with Apache Spark in Azure Synapse Analytics, we also need to normalize data into suitable format before training model.

Apache Spark in Azure Synapse Analytics is one of Microsoft's implementations of Apache Spark in the cloud. Azure Synapse makes it easy to create and configure a serverless Apache Spark pool in Azure. So you can use Spark pools to process your data stored in Azure.

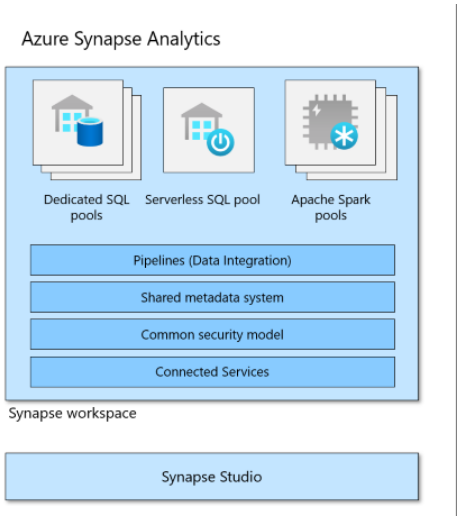


Figure 4.12: Architecture of Azure Synapse Analytics

Apache Spark comes with MLlib, a machine learning library built on top of Spark that you can use from a Spark pool in Azure Synapse Analytics.Spark pools in Azure Synapse Analytics also include Anaconda, a Python distribution with a variety of packages for data science including machine learning. When combined with built-in support for notebooks, it has an environment for creating machine learning applications.

In the application layer, user input the description of property, then encode to a json package and send request to call predict model through provided API. A Json Package of result will be returned after calculate the predicted prices.

5 System implementation

5.1 Crawling Component

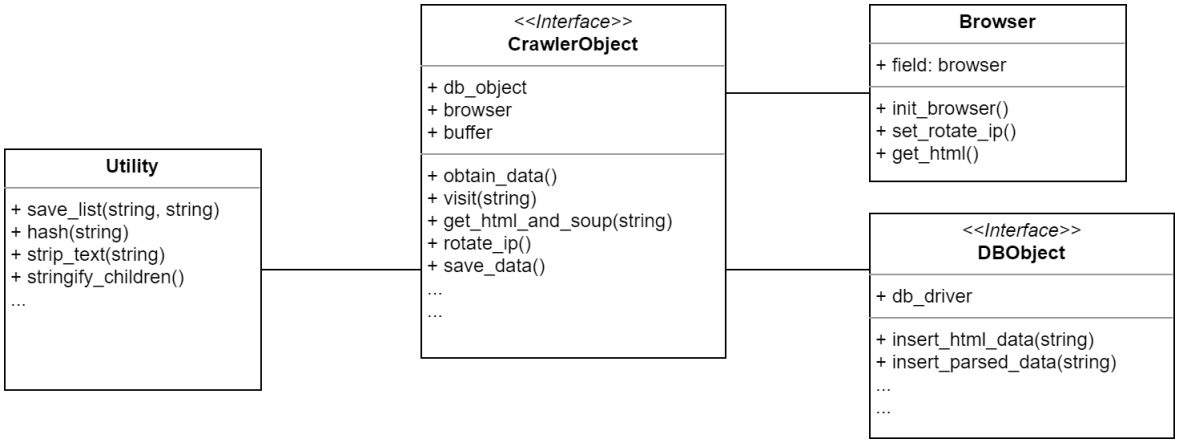


Figure 5.1: Implement of crawling process.

Implementation of Crawling component contains 4 important classes:

- **CrawlerObject:** - CrawlerObject is the main class which is the only place to carry out crawling activity of the system.
- **Brower:** - This class handles Selenium library that supports CrawlerObject to get HTML of web page from URL input.
- **DBObject:** - This class provides API methods that communicates with Cosmos DB library. It is responsible for saving crawled data to the data warehouse.
- **Utility:** - Utility class provides static methods support for the crawling process.

The main class, Crawler Object, contains three main fields:

- *db_object* is an instance of DBObject class.
- *browser* is an instance of Browser class.
- *buffer* is the array to store HTML data in the memory before pushing to Database.

When the beginning of crawling process, the *obtain_data()* is the central method of the class, which is called first. The method *visit(URL)* is to access input URL, request

HTML, then extract to a list of valid URL to access later and save the HTML page source to buffer if the URL is a post. The *rotate_ip()* method is used for change ip address of worker in case of activating **anti-crawling-resistance** mode.

Anti-crawling Resister

To implement Anti-crawling Resister, the system uses web driver to simulate the browser to get HTML and manipulate the web like a normal user. Selenium web driver is a web driver emulator that supports many different types of browsers, cross-platform, supports interaction with web apps with a powerful set of libraries, especially stability and ease of customization.

The Anti-crawling Resister system is implemented together with the data collection system, when the system needs to run in **anti-crawling-resistance** mode, it only needs to send a message to the data crawling system. By default, data crawling components will run in a non-**anti-crawling-resistance** mode.

5.2 Data Parsing Component

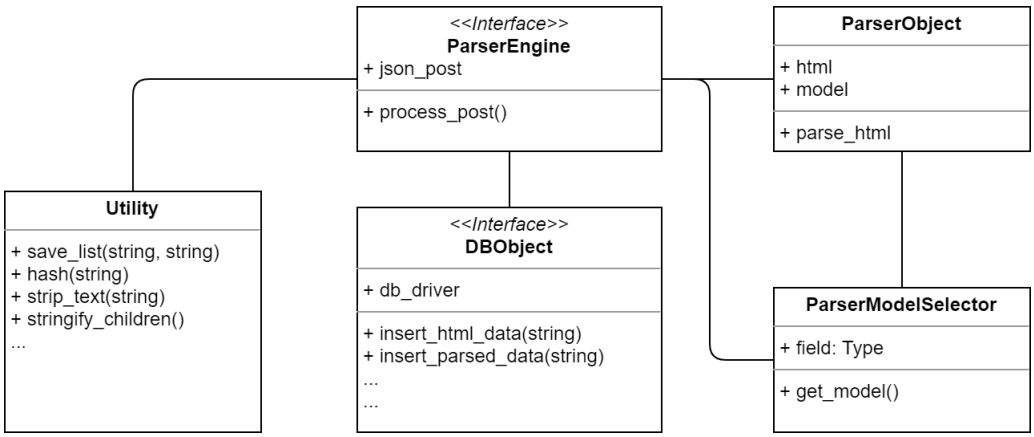


Figure 5.2: Implement of parsing process.

Implementation of Crawling component contains 5 important classes:

- **ParserEngine:** - ParserEngine is the top level class which is the only place to carry out parsing process.
- **ParserObject:** - This class is used in ParserEngine. It is responsible for taking inputs, HTML text and Parser Model, to extract data.

- **ParserModelSelector:** - This class is also used in ParserEngine. It is responsible for select the right Parser Model, as an input for ParserObject.
- **DBObject:** - This class provides API methods that communicates with Cosmos DB library. It is responsible for saving parsed data to the data warehouse.
- **Utility:** - Utility class provides static methods support for the crawling process.

The current extracting component uses two technique: XPath Selector model and Spacy NER model. XPath Selector allows to retrieve the content at one or more nodes in the HTML document. XPath is syntax that define sections of HTML (XML) documents. It use meaningful expressions as the path to navigate to the location to extract information. In order to facilitate the process of extracting data from data collected from many websites, with different interface structures, in the current system, we build common extraction configurations for all sites.

Model Attributes	Type	Description
Features	String	This is the key name of the real-estate property: price, date, tile, address, etc.
Default	String	This is default value of a Feature, in case using xpath obtains null value.
Xpath	String	Providing xpath string that locates value of the feature in HTML text.
Pos_take	Auto	When the Xpath return a list of element, this attributes defines which one to select. There are 3 type of value: • A number: select an element at the index of the number. • A regex string: select an element that matches the regex • null value: merge text of all element in the list.
Regex_take	String	This regex string is used to select a part of the string return by the above filter.
Regex_valid	String	This regex string is to validate the result which must meet any rule.
Len_valid	Integer	This number is to validate the result which must meet string length.
Importance	Integer	The larger number is, the higher the importance of this attribute is. It is used for calculating score of Parser model when parsing any HTML text.

Table 5.1: Attributes of Xpath Selector Model

Filters	Value
Features	address
Default	null.
Xpath	//*[@id='ContentPlaceHolder1_ProductDetail1_divlocation'].
Pos_take	^[Kk]hu [Vv]ực:.*[Tt]ại(.+)$
Regex_take	[Đđ]ường(.+)
Regex_valid	null
Len_valid	10
Importance	3

Table 5.2: Example for Model filters of "address" attribute of nhadat247.com.vn model

5.3 Data Normalizing Component

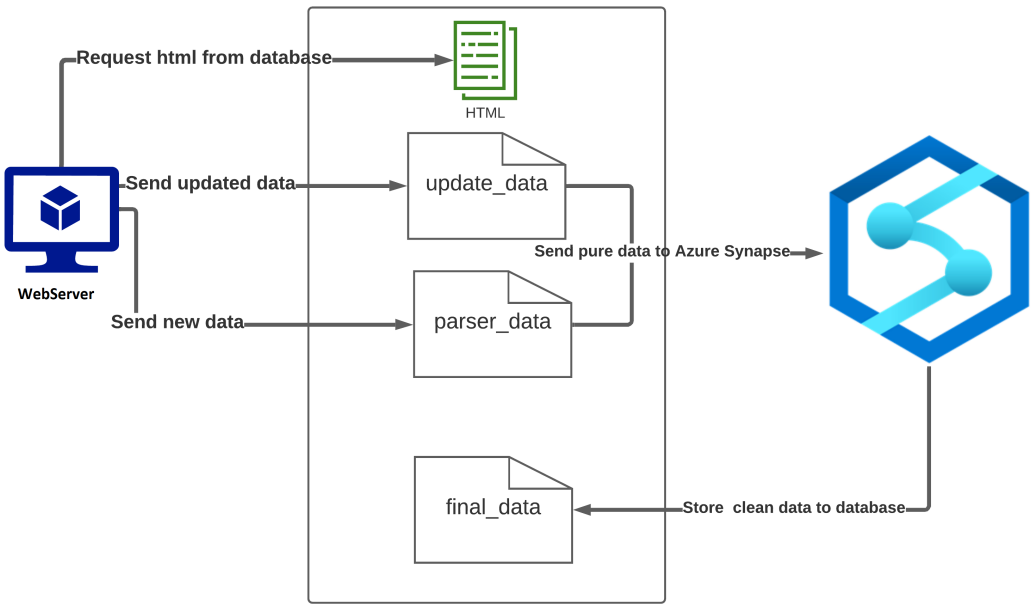


Figure 5.3: Implement of preprocessing component.

The data preprocessing of the system takes the step of normalizing the data and performing outlier removal.In this pre-processing architecture, there are 2 containers which in charge of storing raw data. By using Azure Synapse Link, raw data will be transferred to Spark Pool as a Spark Dataframe to be cleaned and normalized.
This process is done through the following steps:

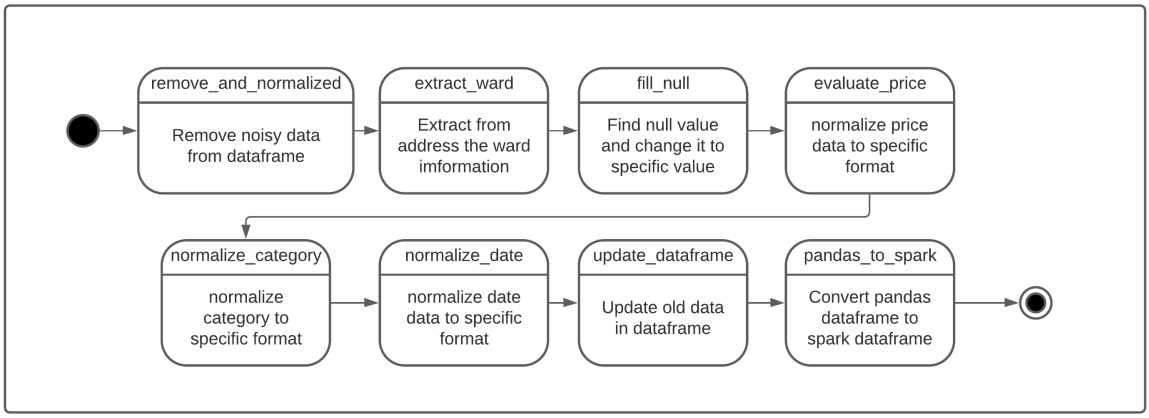


Figure 5.4: Stages of normalizing system.

- **remove_and_normalized**: On e-commerce sites, usually the information in the post is entered by the user. As a result, posts often lack the data that e-commerce sites recommend.For example,user just fill in surface but not width and length,only price but not the surface,address....For those which missing important attribute, we decide to ignore these post and delete them from database.The types of data we decide to remove from the database are:
  - Price unit which include:"Triệu Tỷ","Tỷ/m^2"....With price unit, we only accept Triệu, Tỷ and Triệu/m^2.
  - Number of bedrooms and toilets that not integer.
  - Surface, length, width that not in integer or float...
  - Some location which user forgot to choose: "– Chọn Tỉnh/Thành phố –",....
  - Post that price is null.

  Since each e-commerce site has a different data storage format such as in chotot, region data is saved as "Thành phố Hà Nội" whereas the other is "Hà Nội". In this step, we also change some attributes to a specific concept to handling later:
  - Change format of float data: 5,4 to 5.4..
  - In region, delete all words such as "Thành phố","Tỉnh"
  - In district, delete all words such as "Thành phố", "Quận", "HUyện", "Thị xã"..
  - Some location which user forgot to choose: "– Chọn Tỉnh/Thành phố –",....

  Then return convert it from Spark Dataframe to Pandas dataframe then return.

- **extract_ward**: In parsed data, there is not data of ward, therefore, we need to extract ward data from attribute address. For example:"Đường Phúc Tân, Phúc Tân, Hoàn Kiếm, Hà Nội" to "Phúc Tân".
- **fill_null**: Because when crawl data from real estate e-commerce website, if an attribute is missed, worker will set it value to null. This function will find all value that is null, if it is important data, change to to NoneType object, which will be delete later, else change to specific value.
- **evaluate_price**: After replace the currency unit with specific value, the price has the format:"10*1000000000" for billion unit, "900*1000000" for million unit and "15*45*1000000" for million per square meter. Now we convert it to one format:10, 0.9, 0.675(billion)...
- **normalize_category**: Each real estate e-commerce website has its format of category such as:"chung cư","căn hộ"... We define category into 7 types which are:"nhà","chung cư","đất thổ cư", "đất nền dự án", "đất nông nghiệp", "đất công nghiệp","các loại bất động sản khác".
- **normalize_date**: In some situation, the date of post that crawled has format such as:"hôm nay","hôm qua"....We convert all date to format DD/MM/YY.
- **update_dataframe**: For all data that in the updated_data container, these data will be updated to final_data container following by these url_hash.
- **pandas_to_spark**: Convert Pandas dataframe to Spark dataframe to update and save data to database.

After data is cleansed and normalized, it will be saved back to final_data container through pipelines.

5.4 Storing System Component

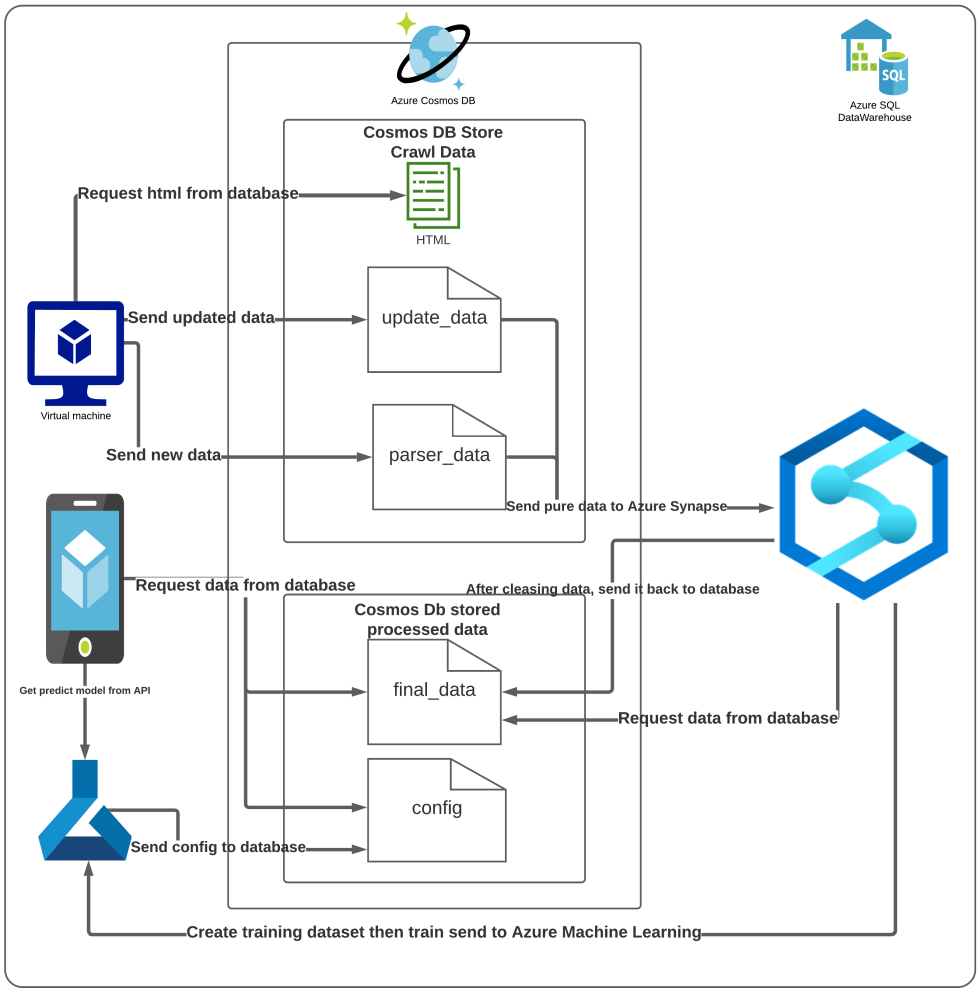


Figure 5.5: Implementation of Storing System

For the storing system component, we divided database into 2 part: **Cosmos DB Stored Crawl Data** and **Cosmos DB stored processed data**. Whereas the first part contains html documents and raw data, second part contains clean data which is used for data mining and serve the application

Cosmos Db Stored Crawl Data

html container: Is a container that stores the html document of the post URLs.

Atribute	Explain	Example
url_hash:key	Key is generated by hashing URL	d34f85b13c20e5351c1336f077ac26f9
url:string	Url of the post	https://batdongsan.com.vn/ban-nha-rieng-duong-pham-van-chieu-phuong-14-8/4-4-ty-50m2-hem-xe-hoi-tranh-14-q-go-vap-pr29908990
crawl_date:string	Data is crawled in that day	19/06/2021
post_date:string	The date the post was posted.	22/05/2021
status:int	If this html document has not parsed, it has value 0. Otherwise, it will turn to 1.	(0,1)
Type:string	Type of data crawl	post
Html:string	Html document stored	"<html><head><script async=\"\" src=\"//stc.za.zaloapp.com/v3/za.js?18797\"></script><script async=\"\" src=\"https://api.ipify.org?format=jsonp&..."

Figure 5.6: Schema of html container.

parser_data and updated_data: There are 2 container to stored parsed data: parser_data, which contains brand new data and updated_data, contains data that exists in final database but need to be updated. Therefore, these schema is the same. With the attribute status, it has initial value equal 0, after parsed by web server, it's value turn into 1.

Atribute	Explain	Example
url_hash:key	Key is generated by hashing URL	d34f85b13c20e5351c1336f077ac26f9
url:string	Url of the post	https://nhadat247.com.vn/ban-can-ho-chung-cu/nha-o-xa-hoi-himlam-thuong-thanh-mo-ban-doi-1-gia-tu-15-trieum2-pr115614.html
crawl_date:string	Data is crawled in that day	19/05/2021
Parser_date:string	Data is parsed in that day	22/05/2021
status_int:int	Item that has been parsed from *HTML container* will has status equal 1.Otherwise it has value equal 0.	(0,1)
Detail:[content:string]	Is where the extracted data is stored.	"detail": { "region": "Hà Nội", "street": "Gia Quất", "district": "Long Biên", "user": "Nguyễn Phượng", "price": "1,5 Triệu /m²", "address": "Gia Quất - Phường Thượng Thanh - Long Biên - Hà Nội", "surface": "55", "bedrooms": "2", ... }

Figure 5.7: Schema of parser and updated container.

Cosmos Db Stored Processed Data

final_data: Is a storage place for processed data from extracted raw data, used as input for data mining processes to provide application information and a knowledge base for data recommendation processes.

Atribute	Explain	Example
url_hash:key	Key is generated by hashing URL.	ba07158c15d4b7bdfac22161da723144
url:string	Url of the post.	https://nha.chotot.com/tp-ho-chi-minh/quan-9/mua-ban-nha-dat/85327386.htm
date:string	The date the post was posted.	22/05/2021
address: string	Address of property.	Đường Số 1, Phường Long Trường, Quận 9, Tp Hồ Chí Minh
surface:string	Square metter of property.	90
Length:string	Length of property.	21
Description:string	Description of property.	Bán nhà đường số 1, P.Long Trường, DT 4,3 X 21 1L giá 4,1 tỷ\nChủ cần tiền bán gấp nhà Hẻm đường số 1. Nhà xây dựng kiên cố 1L được thiết kế 4p.ngũ, 3p.tắm đầy đủ tiện nghi. Khu vực xung quanh yên tĩnh thoáng mát và dân trí cao.\n- Cách đường chính Nguyễn Duy Trinh chỉ 100m\n- Tiện ích xung quanh: trường học các cấp, cản Phú Hữu, gần UBP.Long Trường, phim trường Cabin, bách hoá xanh, thế giới di động\n- Đầy đủ tiện ích xung quanh làm tăng giá trị đột biến BĐS trung khu vực\n- Hẻm 3,5m\n- Hướng TN\n- Pháp lý: Sổ hồng \n- Giá 4,1 tỷ (giá tốt trong khu vực)\n==========\nCÔNG TY TNHH ĐẦU TƯ DỊCH VỤ BĐS LONG ĐIỀN REAL\n - Nhận Ký Gởi Mua Bán Cho Thuê, Dịch Vụ BĐS\n- Hỗ trợ vay vốn ngân hàng từ 60% - 80%.\n- Miễn phí tất cả các dịch vụ tư vấn BĐS",
Ward:string	Name of ward which property located.	Long Trường
Title:string	Title of the post.	Bán nhà đường số 1, P.Long Trường, DT 4,3 X 21 1L
Bedrooms:string	Number of bedroom.	4
Toilets:string	Number of toilet.	3
Phone:string	Phone number of seller.	0906704839
Street:string	Name of Street which property located.	Số 1
Price:float	Price of property.	4.1
Width:string	Width of property.	4
Legal:string	Legal status of the property.	Đã có sổ
Category:string	Type of real estate property.	Nhà
Region:nhà	Region of property	Hồ Chí Minh
User:string	Name of seller	Thanh Nhân
Homepage:string	Website that post crawled from.	Chotot.com

Figure 5.8: Schema of final_data container.

config: Is a storage place for config of predict model.After training model, in some situation, there will be changes in the number of parameter to request model from

server. In addition, there are 4 type of predict model in the system: main, chotot, nhadat, batdongsan..., each model has the different number of parameter.Therefore this is needed to store some information for request model.

Atribute	Explain	Example
Size:int	Number of parameter that needed to call predict price model	484
Model:string	Type of model that user want to call. There are 4 type: main, chotot, nhadat, batdongsan.	Main
List:{parameter: string}	Contains a dictionary of model parameter and its index to call model.	{"surface": 0, "length": 1, "bedrooms": 2, "toilets": 3, "width": 4, "An Giang": 5,.....}

Figure 5.9: Schema of config container.

The system uses the Azure Synapse Link plug-in to create a connection between CosmosDB and Azure Synapse Analytics platform that requires intensive processing.

5.5 Data Visualization Component

With the large amount of data collected, for businesses, the data warehouse system not only analyzes the data but also has to visualize the data in it. The purpose of this is to have a visual and holistic view of the data structure, and to have reports on the data analysis. From there, make specific plans for future development.The topic has created the basis for building a system to meet business requirements so that it can collect, store, and consult the market to make business decisions.

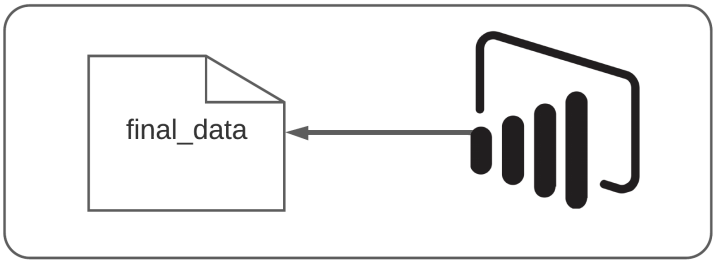


Figure 5.10: Request data for visualization.

In this system, Power BI will connect to final_data container through Cosmos DB

URL and provided connection string.After that, Azure Cosmos DB data can be imported, transformed, created reports, and published the reports to Power BI such as:

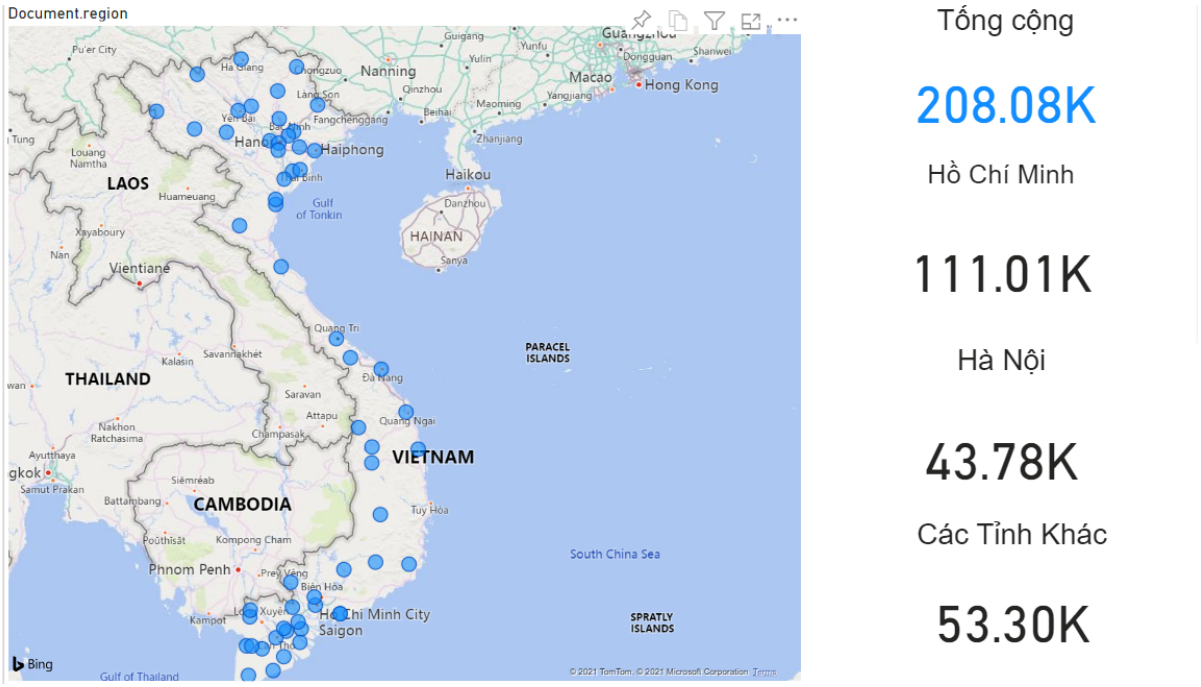


Figure 5.11: Example report 1 for data visualization.

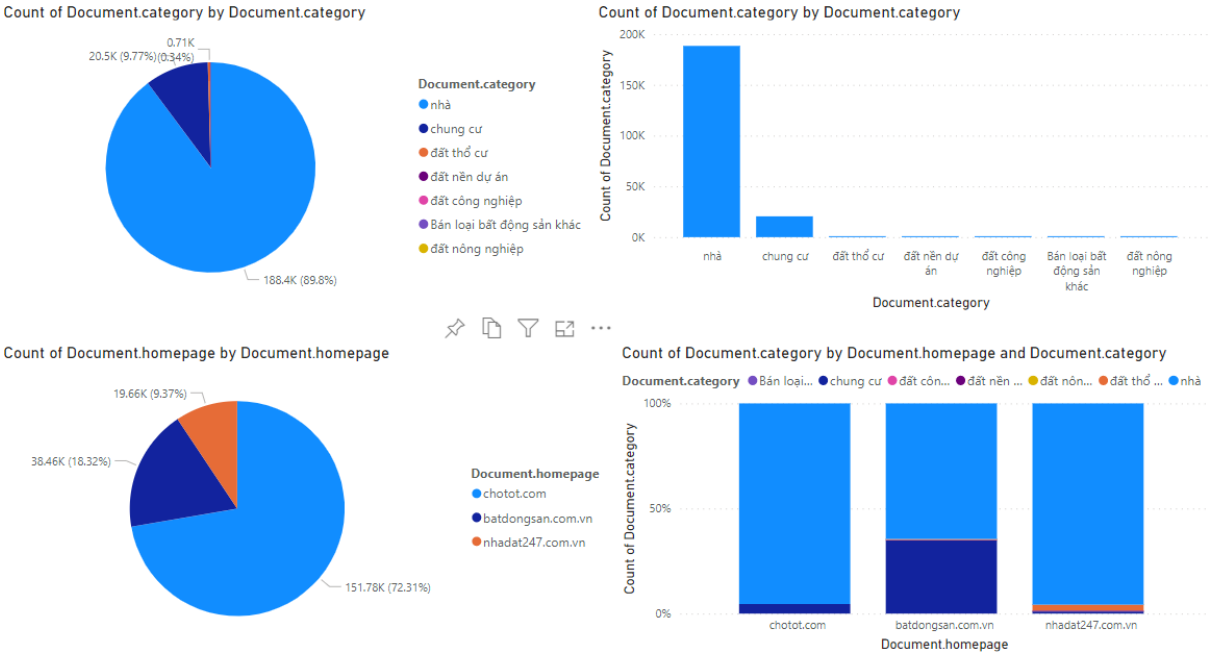


Figure 5.12: Example report 2 for data visualization.

5.6 Prediction System Component

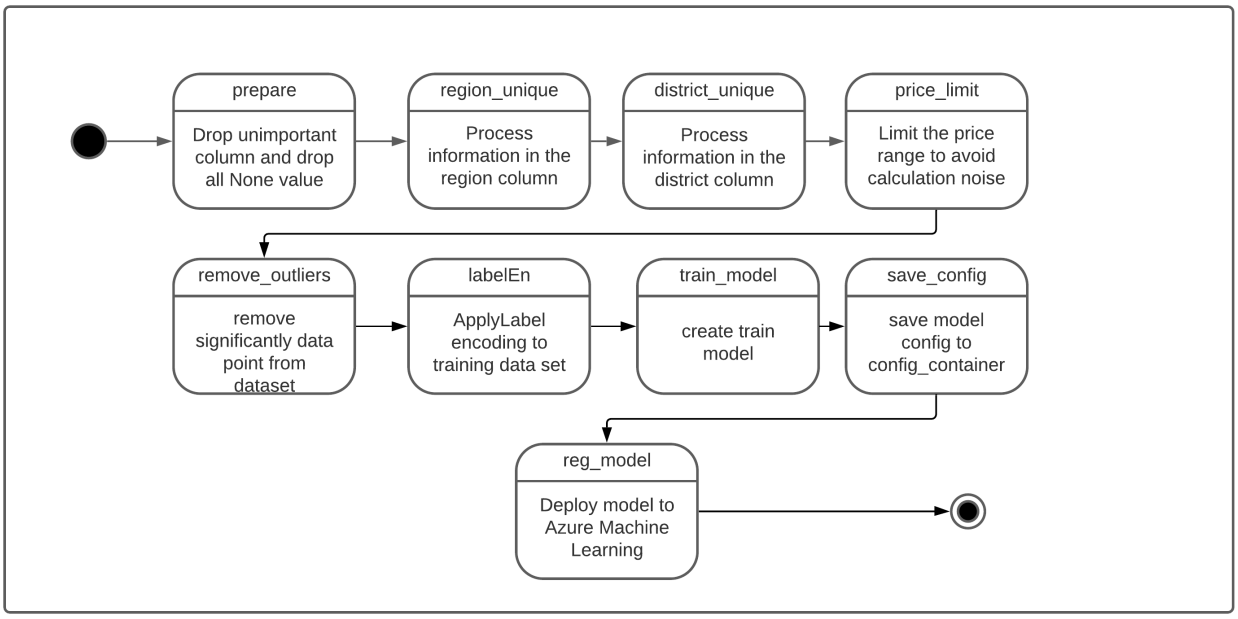


Figure 5.13: Implementation of training prediction model.

By using Azure Synapse Analytics, data from Cosmos is transferred to Spark Pool using pipelines and convert to Spark Dataframe used for model training purposes.This process is done through the following steps:

- **prepare**: The data in dataframe is taken from a container which is used for query task.Therefore it included some columns that are not necessary for training purpose. At first step, this function will drop all the columns that are unimportant and keep important ones which are:
  - surface, length, bedrooms, toilets, price, district width, category, region, homepage..

  Then, Spark will drop all None Type data which as it may cause error in calculation.
- **region_unique**: Calculate the number of item in each region. If a region has less than 10 items, it will be grouped into a group called "other_region".
- **district_unique**: Calculate the number of item in each district. If a district has less than 10 items, it will be grouped into a group called "other_district".
- **price_limit**: Because on the online real estate market, there are homes priced too high or too low. For items, valued at less than 400 million VND or higher than 100

billion, we will remove them from the training dataset.

- **remove_outliers**: remove all data points that is significantly different from other data points in a data set.
- **labelEn**: We use regression for this prediction model. However, there are some columns which these data type are not suitable for input of algorithm. For example, district, region ...are in String data type. Therefore, we decide to apply label encoding to normalize each unique items of these column into value.

	surface	ward	bedrooms	...	homepage	length	width
0	70	10	5	...	nhadat247.com.vn	11	6
1	40	Yên Hòa	4	...	nhadat247.com.vn	10	4
2	46	Phú Đô	4	...	nhadat247.com.vn	11	4
4	42	Đông Hưng Thuận	2	...	chotot.com	7	6
5	30	Hạ Đình	6	...	nhadat247.com.vn	10	3
...	...	...	...	...	...	...	...
226700	91	1	7	...	batdongsan.com.vn	22	4
226706	154	Đông Hội	5	...	batdongsan.com.vn	12	12
226710	64	Tân Thuận Đông	4	...	batdongsan.com.vn	16	4
226713	60	Tân Đông Hiệp	3	...	batdongsan.com.vn	2	25
226715	23	Thanh Xuân Bắc	2	...	batdongsan.com.vn	5	4

	surface	ward	bedrooms	...	homepage	length	width
0	70.0	3	5	...	nhadat247.com.vn	11.0	6.0
2	46.0	728	4	...	nhadat247.com.vn	11.0	4.0
14	199.0	804	4	...	chotot.com	14.0	4.0
15	42.0	30	2	...	nhadat247.com.vn	10.0	4.0
17	87.0	10	7	...	nhadat247.com.vn	14.0	6.0
...	...	...	...	...	...	...	...
226696	71.0	1211	3	...	batdongsan.com.vn	10.0	7.0
226700	91.0	1	7	...	batdongsan.com.vn	22.0	4.0
226706	154.0	1249	5	...	batdongsan.com.vn	12.0	12.0
226713	60.0	1093	3	...	batdongsan.com.vn	2.0	25.0
226715	23.0	868	2	...	batdongsan.com.vn	5.0	4.0

Figure 5.14: Implementation of training prediction model.

- **train_model**: Train model with Linear Regression model from sklearn library in python with this train data set.
- **save_config**: Since the data is updated frequently, the number of parameters of the input prediction model also changes. So, after training the model, we find out the necessary information to use that model and save it to config container for easy use.
- **reg_model**: Finally, we register models to Azure Machine Learning through Spark, then deploy the models to publish for system.

6 System Execution And Evaluation

6.1 System execution

- System of workers:
  - Web server that controls crawling and parsing process.
    - Front-end: React Native.
    - Back-end: Python3.7 Django.
    - RabbitMQ server: distribute tasks to workers.
    - OS: Windows 10.
    - Hardware: IntelCore i5, 4Gb memory.
  - Workers system that perform tasks delivered by server. Include 5 virtual machines.
    - Running on Conda Python.
    - RabbitMQ client: receive task delivered by web server.
    - OS: 5 machines run with Windows Server 2019 Base.
    - Hardware: ECUs, 1 vCPUs Intel Xeon(R) - 2.4 GHz, 1 GiB memory, EBS only.
  - Create small database cluster for data transferring between web server and workers. Using MongoDB.
- Create database in Cosmos DB which include:
  - HTML container for storing html document.
  - parser_data and update_data for storing parsed data from workers.
  - final_data for storing clean data.
  - config for storing recommendation's config.
- Setup Azure Synapse Analytics workspace:
  - Create a Spark Pool medium size with 8vCores/ 64 GB with python 3.6, Apache Spark 2.4 using for processing data.
  - Create a Synapse Link Services to connect CosmosDB through cloud-native HTAP technique.

  - A pre processing data tool kit(a notebook) in Python running on Spark Pool to normalize raw data.
  - A notebook to training prediction model.
  - Setup a pipeline to implement a trigger used to enable data preprocessing to update old data, transfer new data to the database, and update prediction model.
- Set up Azure Machine Learning workspace:
  - Create a Linked Service between Azure Synapse Analytics and Azure Machine Learning.
  - Connect to Spark Pool in Azure Synapse Analytics.
- Set up work space for mobile application:
  - Flutter: Flutter version 2.0.4, Dart version 2.12.2.
  - A test device: An Android Virtual Device (AVD), API level 30, system image R (Android 10.0+).

6.2 Evaluation

6.2.1 Data crawling component

The distributed data collection system is deployed on 5 virtual machines that can be scaled to any computer.

The time for each crawler to collect data for a specific URL in the absence of anti-crawling solution is about 5s and in case of integrated anti-crawling-resistance solution is about 10s.

When the collection components do not integrate anti-crawling solution, there is only 1 website, batdongsan.com.vn, that blocks when accessing the website accessing without browser, so when the use of browser to request URLs will no longer be blocked.

During the processes of data collection or data extraction, if any worker performs a task that is not successful, the system will record it and assign it to another worker when a worker is free to ensure the task is always completed.

For the system that is testing on 5 machines, the system always ensures the data source when needed by other components. The total amount of data that the system

collects is more than 500000 posts, of 3 websites. The average speed when running parallel on 5 machines is one thousand posts per hour.

6.2.2 Data parsing component

The data extraction system operated stably. Currently, the system using Xpath Selector Model which gives a high-precision and basically clean data output.

The data extraction system was tested with 1000 URLs for 2 cases where the organization system on a single machine and the organization system on 5 parallel machines. The results show that when conducted on a single machine, the extraction time is about 12 minutes. When deploying parallel extraction on 5 machines, the time is shortened to about 2 minutes.

6.2.3 Data storage component

- Azure Cosmos DB is used as a storage system on the Azure Synapse Analytics platform.
- **Generality**: The system architecture is designed to handle many different types of data for different requirements.
- **System Scalability**: End-to-end database management, with serverless and automatic scaling matching your application.Serverless model offers spiky workloads automatic and responsive service to manage traffic bursts on demand.Using partitioning to scale individual containers in a database to meet the performance needs of application.
- **System Availability**: Within a region, Azure Cosmos DB maintains four copies of data as replicas within physical partitions. It recommended to setup at least two regions to save data from any failure because at this time data will be replicated in multiple regions, in cosmos DB the container is partitioned horizontally.
- **Minimize maintenance:**: The components in the system are not too complicated to operate because they are guaranteed by the fully-managed serverless system already in the parallel data processing and storage components.
- **Integrated Power BI**:
  - Easy to create report based on data.

  - Auto refresh data and update report daily.
  - Send report to admin using email.
  - Share report workspace through organization.

- Great speed of data ingestion.
- HTAP with Cosmos DB helps transfer data to Azure Synapse for exploration and analytical purposes without affecting storing system. Synapse Link perform powerful analytical queries over application data with no ETL needed.

6.2.4 Decision support system

- User-friendly interface, easy to use.
- The system supports the decision-making of selling prices to easily post real estate for sale.
- We using R_Squared (Coefficient of determination) and Adjusted RSquared to evaluate out prediction models:

```
r2_score of main_model:
0.5432992876938614
Adjusted-R2 :  0.543182846062605
Predicted-R2 0.5428985008219245
*******************************
r2_score of chotot:
0.5103998120632598
Adjusted-R2 :  0.5101841834414178
Predicted-R2 0.5096446477834269
*******************************
r2_score of nhadat247:
0.518224526688406
Adjusted-R2 :  0.5177984998676404
Predicted-R2 0.5163771444767988
*******************************
r2_score of batdongsan:
0.5035964372601679
Adjusted-R2 :  0.5026389214323423
Predicted-R2 0.5004546352344743
```

Figure 6.1: Evaluation of prediction model.

- The system has a fast response time.
- Data is updated regularly.

7 Conclusion

7.1 Achieved result

In this topic, in order to meet the scalability and processing of big data collected from real estate e-commerce websites, we have studied and given an approach to using Azure Big Data technologies as the foundation store, query, and process data. Since then, we have developed a complete and stable data collection and analysis system, including a full range of components for collecting, extracting, preprocessing, storing and analyzing e-commercial data.The components in the system are developed independently of each other, only interacting with each other through common data in the database through connection strings and provided APIs with powerful data recovery mechanisms, ensuring that when one component fails, it won't affect the rest of the components. Moreover, although the components are independent, each part has a separate management mechanism and division of work, but the system still ensures a strict management mechanism, ensuring that the components run stably and without interruption. which one is overloaded or too idle.

In the data crawling component, we focus on getting data from as many sources as possible and also implement policies to prevent anti-crawling from real estate e-commerce. In the first problem, we create an interface for admin which has 4 main functions:add workers into system and test connection,config task for worker to do parsing and crawling, monitor status of current tasks and test parser model. Each e-commerce website has a specific config package which can be edit easily from interface whereas it has changed in format of it's website. In the other problem, although for the 3 websites we crawled, anti-crawling is not yet encountered, but the anti-crawling solution we implemented built into the crawler toolkit and ready to use at any time by simulating normal user behavior while browsing the web by emulating the browser. The data is now only retrieved and fully loaded on the respective browser, instead of directly by processes running in the background.

We have researched and decided to implement the system on Azure's ecosystem with the main components being Azure CosmosDb, Azure Synapse Analytics and Power BI that brings together enterprise data warehousing and Big Data analytics.In addition to having big data stored and processed quickly and efficiently, Azure also provides solutions for data security and data recovery through regularly updated and archived backups.

Hosting on a serverless system helps reduce the complexity of ETL job management as well as maintenance, avoids overload, and divides data streams appropriately.

In addition, for the collected data, we also use it to develop an application to support users in making decisions about real estate valuation, besides giving users comparisons based on real estate valuation. data from one or more different sources.The app has three main features: it allows the user to see the suggested price for an incoming property, it allows viewing of comparable homes and finally it allows to view comparisons of property information. real estate is suggested intuitively

7.2 Evaluate the significance of the topic

The topic has created the basis for building a system to meet business requirements so that it can collect, store, and consult the market to make business decisions:

- Update data from multiple sources.
- High query options, scalable and easy to management according to business purposes.
- The response to system queries is fast even though the data is very large.
- Visualization is done to help businesses have an overall view, easy management and help make future plans.

In addition, developing applications based on many data sources that the system collects is a solution to serve the needs of users in buying and selling real estate when there are many C2C e-commerce sites. in the market makes it difficult for users to compare and make decisions.

7.3 Future development

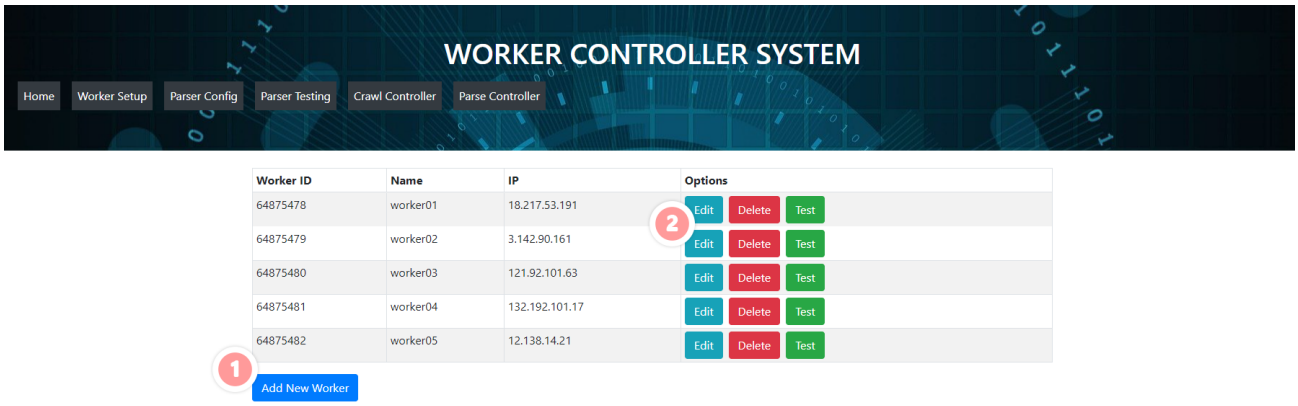
- Develop a behavioral data collector to deceive web servers, make it possible to collect data through the policies of e-commerce websites, and prevent anti-crawling or anti-crawling mechanisms. unexpected change in the structure of e-commerce websites
- Perform data crawling not only from 3 proposed sites but all e-commerce sites in the real estate sector in Vietnam.

- Improve the extraction of information that does not follow a defined structure, depending on the semantics of the poster so that the system can allow businesses to query more useful and accurate information.In addition, fully develop partial update in cosmos DB (which is now only published in private review of Microsoft) to lowers end-to end-latency and can significant reduce network payload
- Improved algorithm to predict house prices more accurate, besides developing more functions to help users compare and make more decisions based on the large amount of data we have.

Appendix: User Manual

Web Admin Server

Worker Setup

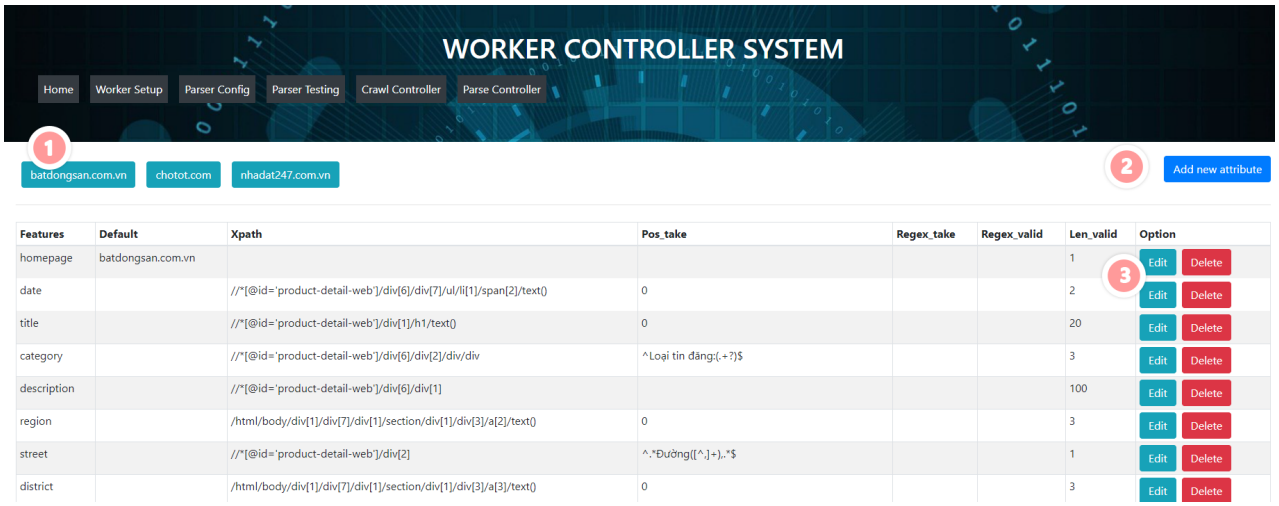


Worker Setup allows admin to add new workers to the system, the information that admin has to provide is a set of: public IP of the workers machine, RabbitMQ client name and password. After adding new workers, admin can test the connections and modifies or even delete the information of workers.

1. Click on Add New Worker, then a form will pop up, which allows admin to fill the information of the worker.

2. There are three Option buttons on each added workers. The Edit button is for modifying, the Delete button is for removing, and the Test button is for testing the connections of RabbitMQ.

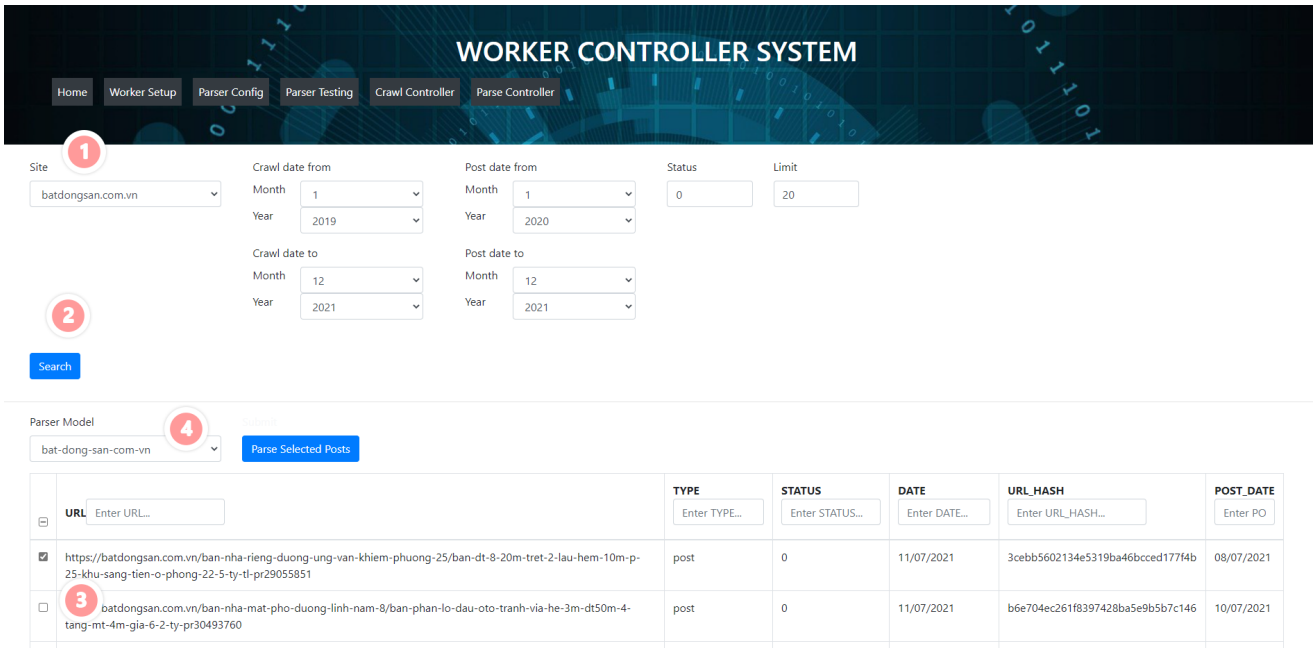
Parser Config



In the data parsing component, we use two types of model: the XPath selector model and the Spacy NER model. However, the XPath selector model is the only one to be modified.

1. Firstly, admin has to select a model to edit.
2. Secondly, admin can add new attributes to the selected model. When a new attribute is added correctly, the data parser is able to using the model to extract this attribute from HTML source code.
3. Moreover, there are two options for editing and deleting saved attributes in the model.

Parser Testing



Parser Testing is a tool which is designed to test the correctness of parser model on the HTML data set.

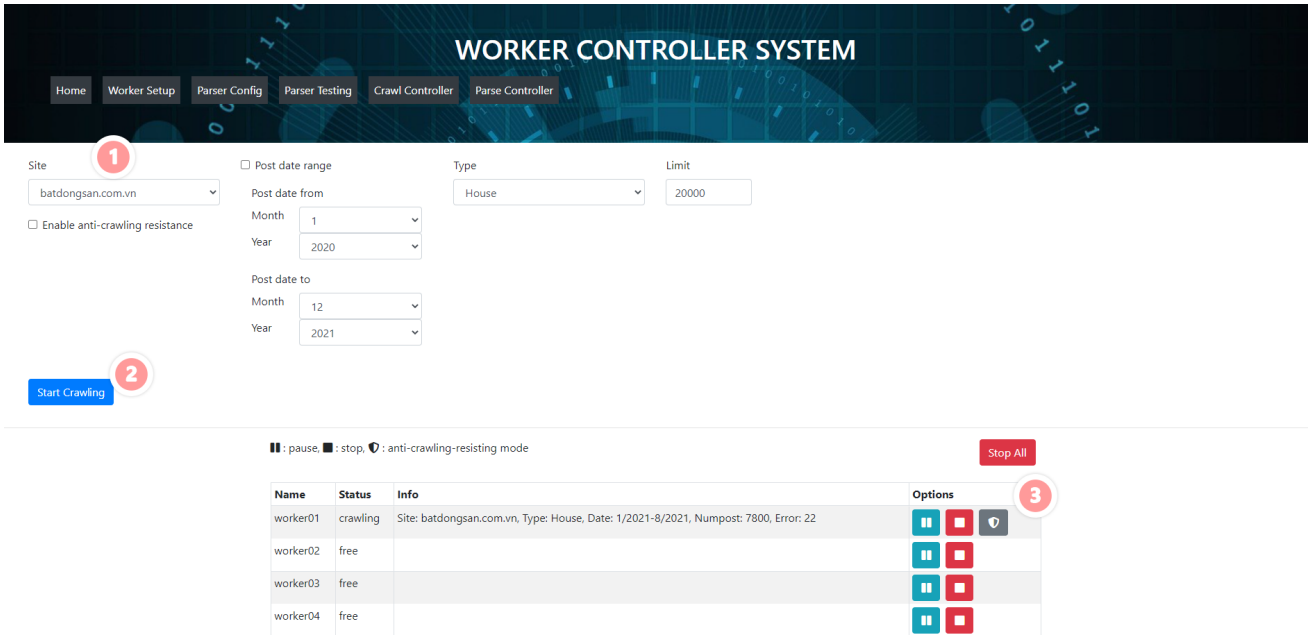
1. Firstly, to start testing model, admin has to search for HTML data record in the data ware house of the system by choosing the search filters.

2. The admin clicks on Search button to obtain data.

3. Admin can select records to do parsing on.

4. Admin selects a model to be test, then click on Test Parser Model button.

Crawl Controller



Crawl Controller is the panel where admin configures and start data crawling processes.

1. Firstly, admin configures necessary parameters:

   - Website: batdongsan.com.vn, chotot.vn, etc.
   - Activate/deactivate anti-crawling resistance mode.
   - Range of the posting date: from 01/2021 to 08/2021.
   - Type of post: house, apartment, etc.
   - Limit number of post.

2. To start crawling, click on Start Crawling button.

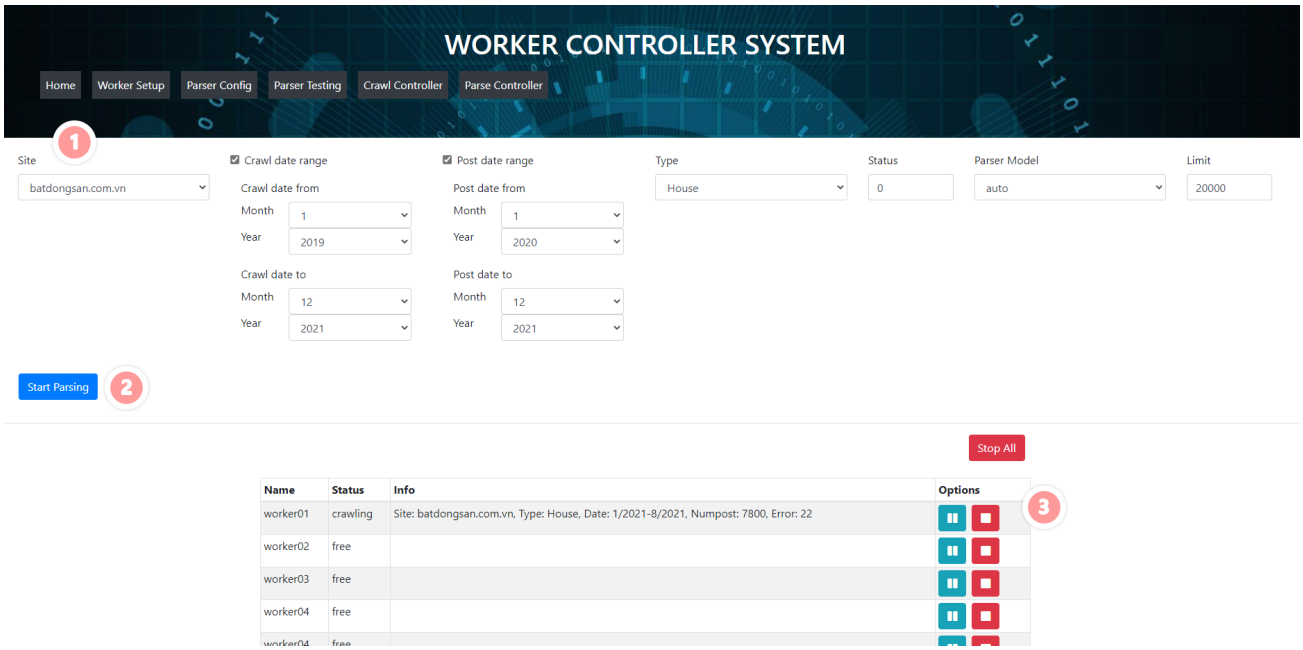
3. There are four option buttons for controlling the running workers.

   – Pause: temporarily postpone running task of a worker, after that admin can resume the task.

   – Cancel: destroy running task of any worker.

– Shield: toggle the activation of anti-crawling resistance mode if worker is crawling.
– Stop all: destroy all of running tasks of all workers.

Parse Controller



Parse Controller is the panel where admin configures and start data parsing processes.

1. Firstly, admin configures necessary parameters:

   - Website: batdongsan.com.vn, chotot.vn, etc.
   - Activate/deactivate anti-crawling resistance mode.
   - Range of the posting date: from 01/2021 to 08/2021.
   - Range of the crawling date: from 07/2021 to 08/2021.
   - Type of post: house, apartment, etc.
   - Status of HTML data record which will increase 1 unit after being parsed completely.
   - Parser Model to extract HTML data.
   - Limit number of post to be processed.

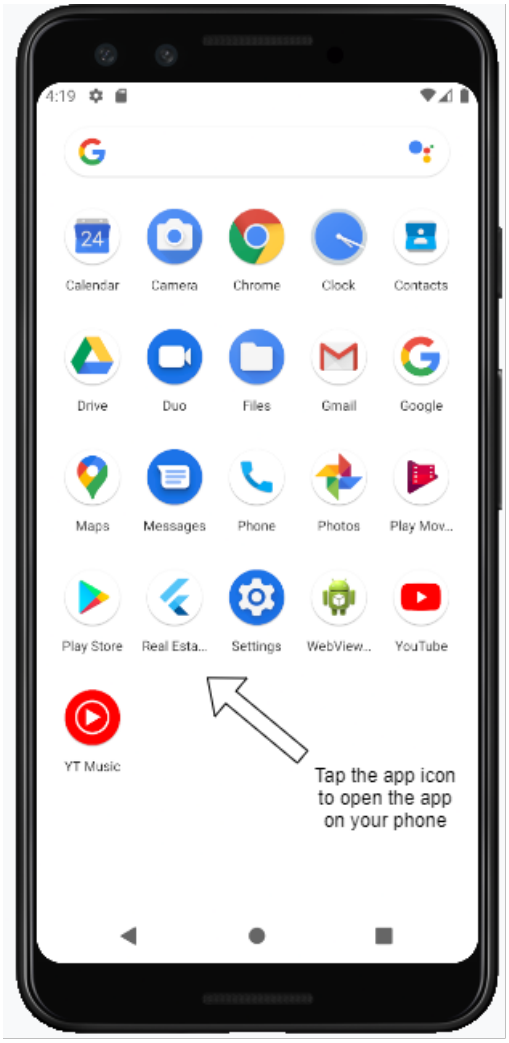
2. To start parsing, click on Start Parsing button.

3. There are four option buttons for controlling the running workers.

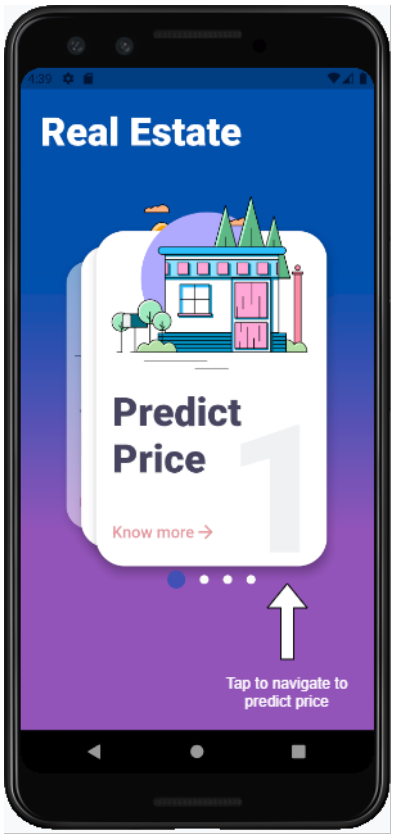
   – Pause: temporarily postpone running task of a worker, after that admin can resume the task.
   – Cancel: destroy running task of any worker.
   – Stop all: destroy all of running tasks of all workers.

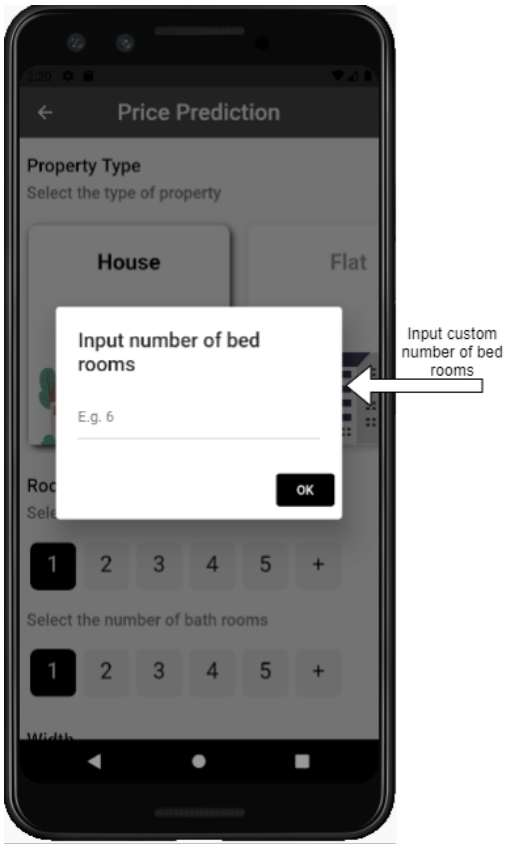
Mobile Application

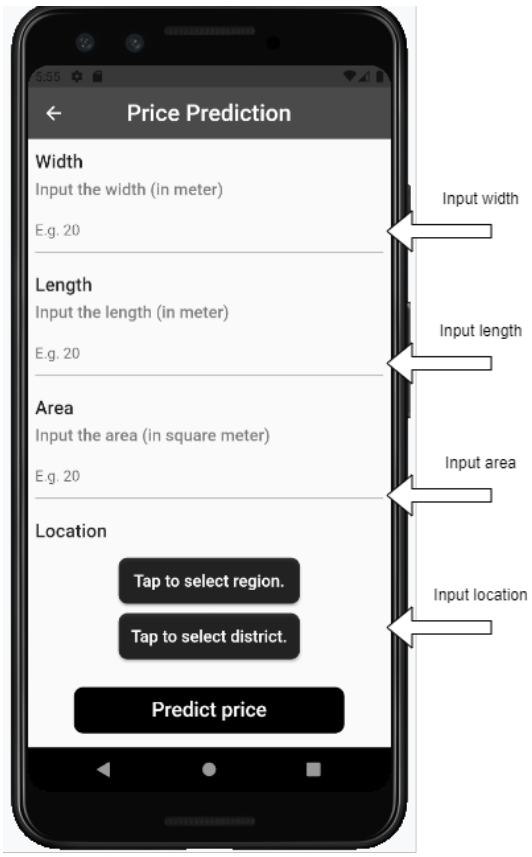
- Get the application depend on your mobile platform: Google Play if you are using Android or App Store if you are using iOS.
- Tap the app icon to open the app on your phone.



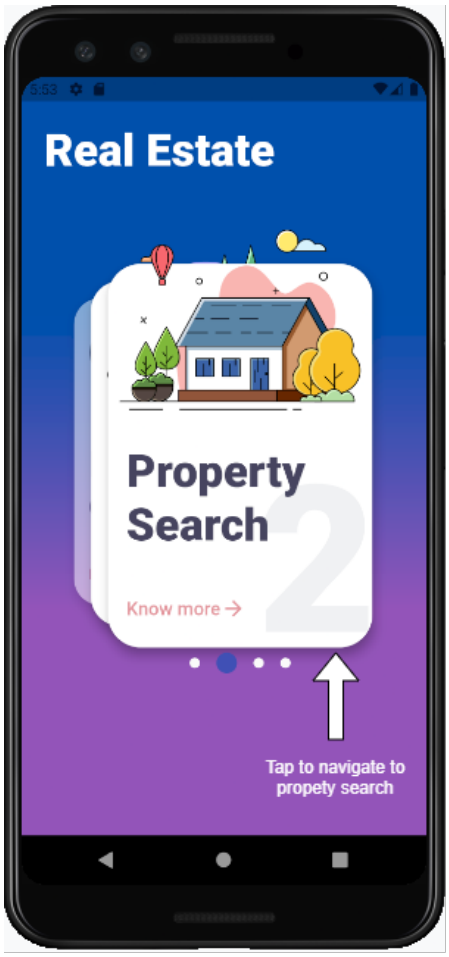
- When the app is loaded, select a functionality you want to execute.
- If you want to predict selling price of your property, tap on the menu and enter description of the property:

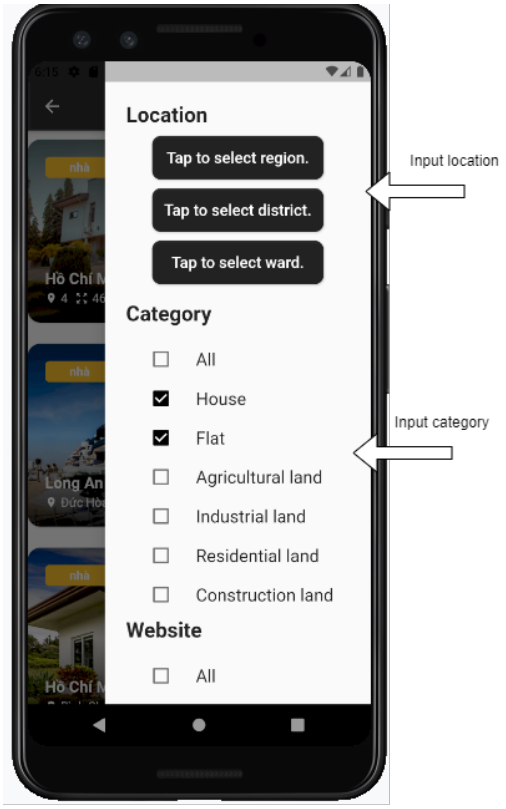


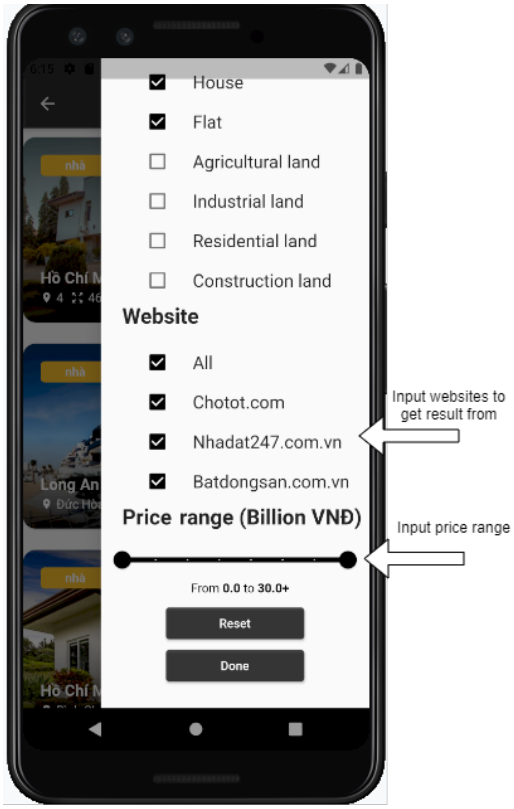




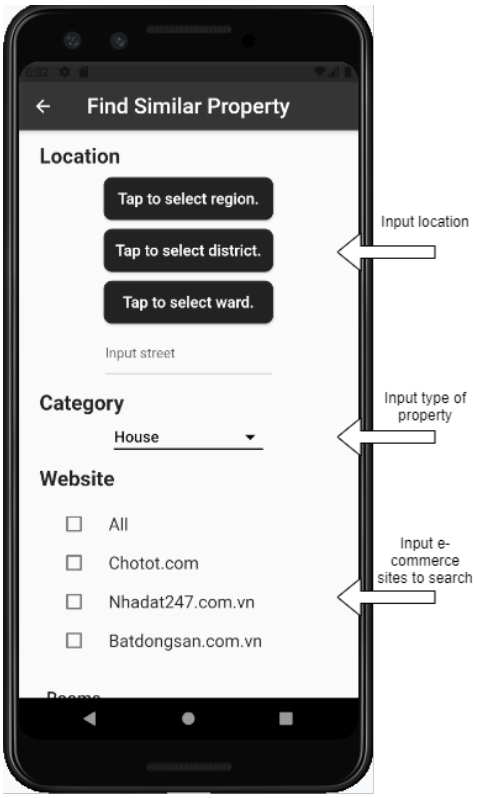
- If you want to search for sale posts, tap on the menu and enter description of the property:

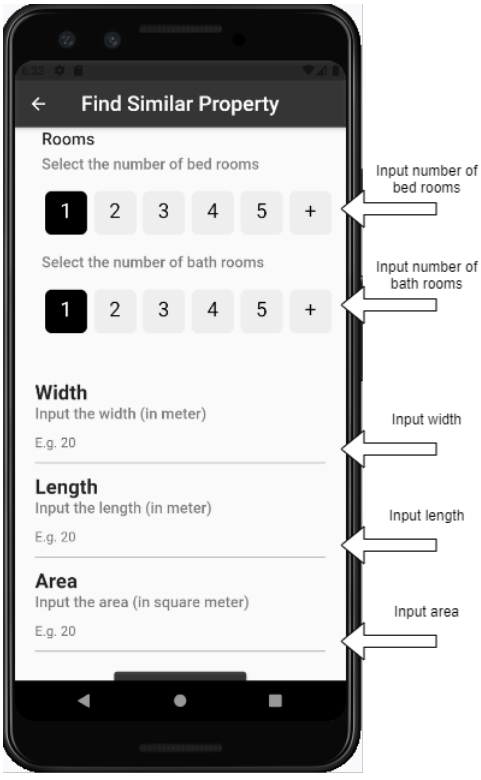

Real Estate
Property Search
Know more
Tap to navigate to propety search


Location
Tap to select region.
Tap to select district.
Tap to select ward.
Category
All
House
Flat
Agricultural land
Industrial land
Residential land
Construction land
Website
All
Input location
Input category

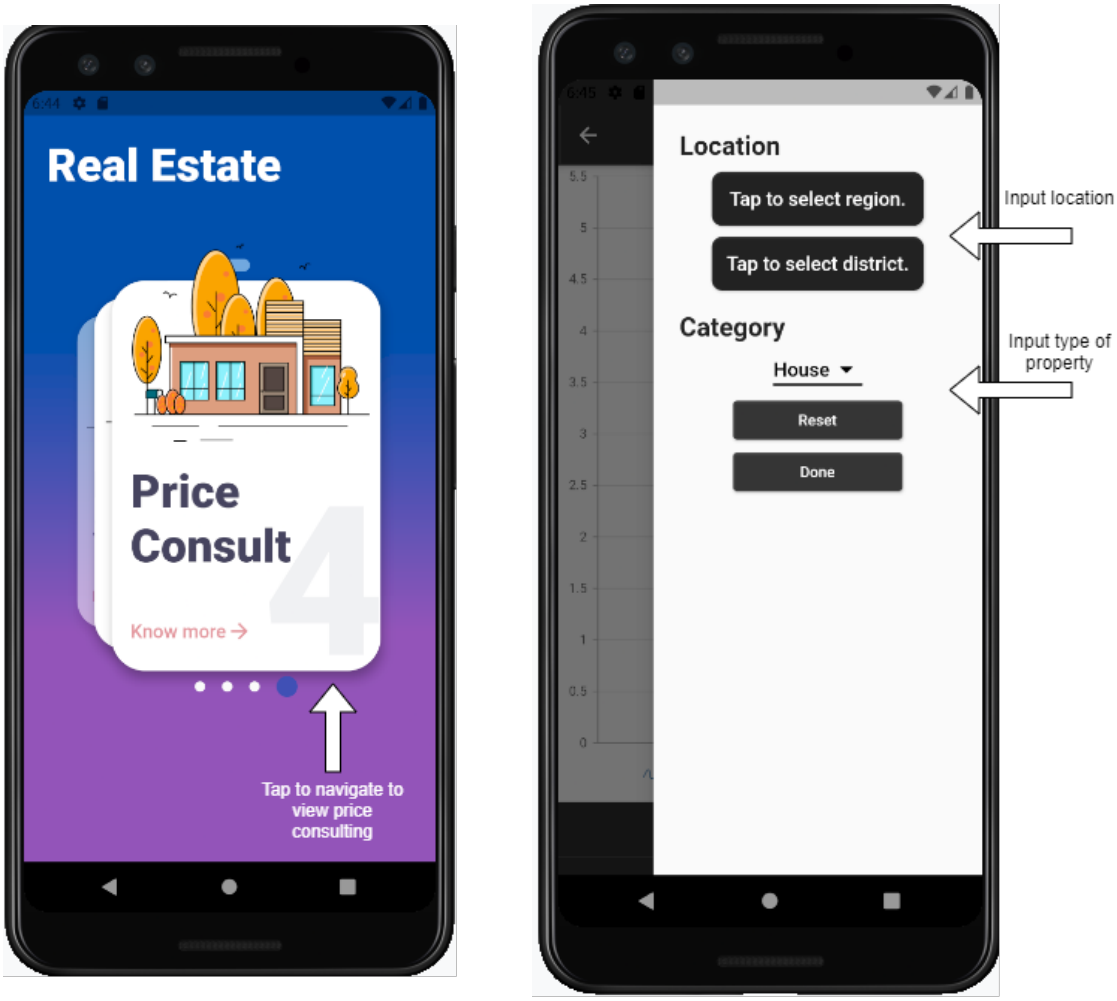

House
Flat
Agricultural land
Industrial land
Residential land
Construction land
Website
All
Chotot.com
Nhadat247.com.vn
Batdongsan.com.vn
Price range (Billion VNĐ)
From 0.0 to 30.0+
Reset
Done
Input websites to get result from
Input price range

- If you want to find similar properties, tap on the menu and enter description of the property:





- If you want to view price fluctuation, tap on the menu and enter some information:



References

1. Introduction to Azure Cosmos DB. https://docs.microsoft.com/en-us/azure/cosmos-db/introduction. Accessed: 2021/06/20.

2. B. Thakur, M. Mann, “Data Mining for Big Data: A Review,” International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, No. 5, pp. 469-473, 2014.

3. U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From Data Mining to Knowledge Discovery in Database,” AI Magazine, Vol.17,pp. 37-54, 1996.

4. M. Kantardzic, “Data Mining: Concepts, Models, Methods and Algorithms,” John Wiley & Sons, Inc., 2002.

5. F. Gorunescu, “Data Mining Concepts, Models and Techniques,” Vol.12, Springer-Verlag Berlin Heidelberg, 2011.

6. Yunus Koloğlu, Hasan Birinci, Sevde Ilgaz Kanalmaz, Burhan Özyılmaz, "A Multiple Linear Regression Approach For Estimating the Market Value of Football Players in Forward Position":n. pag. Web

7. Ozgur, Ceyhun; Hughes, Zachariah; Rogers, Grace; and Parveen, Sufia (2016),"Multiple Linear Regression Applications in Real Estate Pricing". Business Faculty Publications. 61.:n. pag. Web

8. SACHIN KAMLEY, SHAILESH JALOREE, R. S. THAKUR, "MULTIPLE REGRESSION: A DATA MINING APPROACH FOR PREDICTING THE STOCK MARKET TRENDS BASED ON OPEN, CLOSE AND HIGH PRICE OF THE MONTH"(2013),International Journal of Computer Science Engineering and Information Technology Research:n. pag. Web

9. GABDA, DARMESAH; JUBOK, ZAINODIN HJ; BUDIN, KAMSIA; HASSAN, SURIANI, "MULTIPLE LINEAR REGRESSION IN FORECASTING THE NUMBER OF ASTHMATICS", school of Science and Technology University Malaysia Sabah: n. pag. Web

10. A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning. https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/. Accessed: 2021/08/05.

11. What is Gradient Boosting and how is it different from AdaBoost?. https://www.mygreatlearning.com/blog/gradient-boosting/#sh5. Accessed: 2021/08/05.

12. Alexey Natekin and Alois Knoll, “Gradient boosting machines, a tutorial” Department of Informatics, Technical University Munich, Garching, Munich, Germany, 2013.

13. How to Develop a Light Gradient Boosted Machine (LightGBM) Ensemble https://machinelearningmastery.com/light-gradient-boosted-machine-lightgbm-ensemble/. Accessed: 2021/08/05.

14. Understanding Gradient Boosting Machines https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab. Accessed: 2021/08/05.

15. LightGBM (Light Gradient Boosting Machine) https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/. Accessed: 2021/08/05.

16. Y. Ning, X. Zhu, S. Zhu, and Y. Zhang, "Surface EMG Decomposition Based on K-means Clustering and Convolution Kernel Compensation," IEEE J. of Biomedical and Health Informatics, Vol.19, pp. 471-477, 2015.

17. Dvijesh Bhatt, Daiwat Amit Vyas and Sharnil Pandya.(2020). "Focused Web Crawler", Vol.2, 2015 pp. 1-6. Institute of Technology, Nirma University.

18. Pankaj Jainani, "Azure Synapse Analytics — Introduction",22 Feb 2021

19. Enterprise Data Warehouse Architecture https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/enterprise-data-warehouse. Accessed: 2020/06/20

20. What is Azure Synapse Link for Azure Cosmos DB? https://docs.microsoft.com/en-us/azure/cosmos-db/synapse-link. Accessed: 2021/06/20.

21. batdongsan.com.vn, https://batdongsan.com.vn/tin-thi-truong/luot-quan-tam-bat-dong-san-tang-gap-doi-trong-quy-1-2021-ar106434,23/03/2021 09:58

22. Apache Spark in Azure Synapse Analytics https://docs.microsoft.com/en-us/azure/synapse-analytics/spark/apache-spark-overview.

23. Ferrara, E., De Meo, P., Fiumara, G., Baumgartner, R. (2014). "Web data extraction, applications and techniques: A survey. Knowledge-Based Systems", 70, 301–323. doi:10.1016/j.knosys.2014.07.007

24. M. Chau, H. Huy, A. Khuong, D. Dan. (2017) . "Hệ thống thu thập và phân tích dữ liệu thị trường thương mại điện tử Việt Nam". Vietnam.

25. MUSTAFA EL-MASRY . "Azure Cosmos DB High Availability and Disaster Recovery Architecture". JULY 28, 2020

26. Real Time Analytics on Big Data Architecture https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/real-time-analytics.

27. Leonard . "Data Modeling and Partitioning for Relational Workloads". June 30th, 2020

28. How Big Data Helps in Real Estate Analysis. https://www.getsmarter.com. July 16, 2019.

29. Gabriel Morgan Asaftei, Sudeep Doshi, John Means, and Aditya Sanghvi. "Getting ahead of the market: How big data is transforming real estate". October 8, 2018

30. Real Estate Comps: How to Find Comparables for Real Estate. https://www.zillow.com. Accessed: 2021/06/20.