

UNIVERSITY OF ECONOMICS AND LAW

**FACULTY OF INFORMATION SYSTEMS**



**FINAL PROJECT REPORT**

**COURSE: DEEP LEARNING**

**Topic:**

**GROCERY APPLICATION**

**INTEGRATED MACHINE LEARNING MODEL**

**IN REVIEWS SENTIMENT**

*Ho Chi Minh City, November 17, 2023*

UNIVERSITY OF ECONOMICS AND LAW

**FACULTY OF INFORMATION SYSTEMS**



**FINAL PROJECT REPORT**

**COURSE: DEEP LEARNING**

**Topic:**

**GROCERY APPLICATION**

**INTEGRATED MACHINE LEARNING MODEL**

**IN REVIEWS SENTIMENT**

| **Group 5:** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **NO.** | **Name** | **ID** | | **Faculty** | **Role** | **Phone** | **Email** |
| 1 | Vu Van Son | | K204160669 | FIS | Member | 0944244857 | sonvv20416c  @st.uel.edu.vn |
| 2 | Chau Nguyen Huong Thao | | K204162000 | FIS | Member | 0919772802 | thaocnh20416c  @st.uel.edu.vn |
| 3 | Nguyen Tuong Vy | | K204162009 | FIS | Leader | 0764187037 | vynt20416c  @st.uel.edu.vn |

Lecturer: **Ph.D. Tran Duy Thanh.**

*Ho Chi Minh City, August 7, 2023*

# ACKNOWLEDGEMENTS

During this project, we received a significant amount of dedicated assistance and guidance from our lecturer - Ph.D. Tran Duy Thanh and classmates. Thanks to their contributions, we were able to complete our final project.

Despite our best efforts, mistakes may have been made, and we welcome any feedback or constructive criticism you may have to help improve the quality of our project. Your input will serve as a tremendous source of motivation as we continue to develop our project in the future.

# COMMITMENT

We assure that our final project is original and based on research developed by the entire team. In the event that any provided information is found to be inaccurate or misleading, we take full responsibility for any potential consequences that may arise, and we will collaborate closely with our lecturer to rectify the situation.

Ho Chi Minh City, November 17, 2023

**Committed by**

**Group 5**

# TABLE OF CONTENT

[**ACKNOWLEDGEMENTS I**](#_heading=h.23ckvvd)

[**COMMITMENT II**](#_heading=h.ihv636)

[**TABLE OF CONTENT III**](#_heading=h.1664s55)

[**LIST OF FIGURES V**](#_heading=h.3q5sasy)

[**LIST OF TABLES VI**](#_heading=h.25b2l0r)

[**LIST OF ACRONYMS VII**](#_heading=h.kgcv8k)

[**CHAPTER 1: OVERVIEW 1**](#_heading=h.34g0dwd)

[**1.1**](#_heading=h.1jlao46) **Objectives of the project 1**

[**1.2**](#_heading=h.43ky6rz) **Structure of the project report 1**

[**1.3**](#_heading=h.2iq8gzs) **The project implementation plan 1**

[**CHAPTER 2: THEORETICAL BASIS 3**](#_heading=h.3hv69ve)

[**2.1**](#_heading=h.1x0gk37) **.NET Framework 3**

[2.1.1](#_heading=h.1baon6m) Window Form 4

[2.1.2](#_heading=h.3vac5uf) LINQ 5

[**2.2**](#_heading=h.2afmg28) **ML.NET Framework 5**

[2.2.1. Automated Machine Learning 6](#_heading=h.39kk8xu)

[**2.3 Recommendation system 7**](#_heading=h.1opuj5n)

[**2.4. Sentiment Classification 7**](#_heading=h.48pi1tg)

[**CHAPTER 3: METHODOLOGY FOR A RECOMMENDATION MODEL 9**](#_heading=h.2nusc19)

[**3.1**](#_heading=h.1302m92) **Database structure 9**

[**3.1.1**](#_heading=h.2250f4o) **Reviews table 9**

[**3.1.2**](#_heading=h.319y80a) **Partners table 10**

[**3.1.3**](#_heading=h.40ew0vw) **Destinations table 10**

[**3.1.4**](#_heading=h.upglbi) **Bookings table 11**

[**3.1.5**](#_heading=h.1tuee74) **Customers Table 11**

[**3.1.6**](#_heading=h.2szc72q) **Admins table 12**

[**3.2**](#_heading=h.3s49zyc) **Recommendation model 12**

[**CHAPTER 4: EXPERIMENTAL APPLICATION 14**](#_heading=h.meukdy)

[**4.1**](#_heading=h.1ljsd9k) **Function: Login 15**

[4.1.1](#_heading=h.45jfvxd) DFD 16

[4.1.2](#_heading=h.3jtnz0s) BPMN 18

[4.1.3](#_heading=h.4iylrwe) Mock-ups 18

[**4.2**](#_heading=h.2y3w247) **Function: Booking 19**

[4.2.1](#_heading=h.1d96cc0) DFD 19

[4.2.2](#_heading=h.uhjn8eu3veqm) BPMN 21

[4.2.3](#_heading=h.4anzqyu) Mock-ups 23

[**4.3**](#_heading=h.2pta16n) **Function: Reviews management 26**

[4.3.1](#_heading=h.14ykbeg) DFD 26

[4.3.2](#_heading=h.j8sehv) BPMN 28

[4.3.3](#_heading=h.1idq7dh) Mock-ups 29

[**4.4**](#_heading=h.42ddq1a) **Function: Destination recommendations 29**

[4.4.1](#_heading=h.2hio093) DFD 29

[4.4.2](#_heading=h.1vsw3ci) BPMN 31

[4.4.3](#_heading=h.2uxtw84) Application: 31

[**CHAPTER 5: CONCLUSIONS AND FUTURE WORKS 36**](#_heading=h.1a346fx)

[**5.1**](#_heading=h.3u2rp3q) **Results 36**

[**5.2. Limitations 36**](#_heading=h.2981zbj)

[**5.3. Future works 36**](#_heading=h.odc9jc)

[**REFERENCES 38**](#_heading=h.38czs75)

[**APPENDIX 39**](#_heading=h.11si5id)

# LIST OF FIGURES

[Figure 1: The project implementation plan 2](#_heading=h.xvir7l)

[Figure 2: .NET Framework (source: www.extranerds.com ) 3](#_heading=h.4h042r0)

[Figure 3: History of .NET Framework (source: world.optimizely.com) 4](#_heading=h.2w5ecyt)

[Figure 4: ML.NET Framework (source: https://en.wikipedia.org/wiki/ML.NET) 6](#_heading=h.pkwqa1)

[Figure 5: Database diagram 8](#_heading=h.3mzq4wv)

[Figure 6: Diagram for experiment progress of Recommendation system. 11](#_heading=h.279ka65)

[Figure 7: Use case of Travel booking application 13](#_heading=h.36ei31r)

[Figure 8: Data Flow Diagram Level 0 for the Login Function 15](#_heading=h.2koq656)

[Figure 9: Data Flow Diagram Level 1 for the Login Function 16](#_heading=h.zu0gcz)

[Figure 10: BPMN for Login function 17](#_heading=h.1yyy98l)

[Figure 11: DFD level 0 for the Booking function 18](#_heading=h.2ce457m)

[Figure 12: DFD level 1 for the Booking function 20](#_heading=h.rjefff)

[Figure 13: BPMN for Booking function 21](#_heading=h.1qoc8b1)

[Figure 14: DFD level 0 for the Reviews management function 25](#_heading=h.3oy7u29)

[Figure 15: DFD level 1 for the Review management function 27](#_heading=h.243i4a2)

[Figure 16: BPMN for the Reviews management function 27](#_heading=h.338fx5o)

[Figure 17: DFD level 0 for Destination recommendation function 28](#_heading=h.wnyagw)

[Figure 18: DFD level 1 for Destination recommendation function 29](#_heading=h.3gnlt4p)

[Figure 19: BPMN for Destination recommendation function 30](#_heading=h.4fsjm0b)

# LIST OF TABLES

[Table 1. Reviews table 8](#_heading=h.haapch)

[Table 2: Partners table 9](#_heading=h.1gf8i83)

[Table 3: Destinations table 9](#_heading=h.2fk6b3p)

[Table 4: Bookings table 10](#_heading=h.3ep43zb)

[Table 5: Customers Table 10](#_heading=h.4du1wux)

[Table 6: Admins table 11](#_heading=h.184mhaj)

# LIST OF ACRONYMS

| **NO.** | **ACRONYMS** | **MEANINGS** |
| --- | --- | --- |
| 1 | LINQ | Language Integrated Query |
| 2 | CLR | Common Language Runtime |
| 3 | XML | eXtensible Markup Language |
| 4 | UFP | Universal Windows Platform |
| 5 | CRUD | Create, Read, Update, Delete |
| 6 | BPMN | Business Process Modeling Notation |
| 7 | DFD | Data Flow Diagram |
| 8 | GDI+ | Graphics Device Interface |

# CHAPTER 1: OVERVIEW

## 1.1 Objectives of the project

The primary objective of this project is to apply general machine learning models and specifically the recommendation model for text data mining using the ML.NET technology. The subsequent goal is to comprehend the process of constructing an interactive database that interfaces with the software through LINQ technology. To be more specific, the aim of this project is to deploy a simple application using the C# language through the .NET Framework, centered around the travel booking topic. This application will encompass key functionalities such as user login, booking, customer reviews, and recommendation destinations for users to choose from. The ultimate purpose is to reinforce programming skills using the C# language, and simultaneously enhance design thinking by designing the software's user interface.

## 1.2 Structure of the project report

The structure of the report consists of five chapters. **Chapter 1** provides an overview of the project, while the theoretical basis will be presented in **Chapter 2**. **Chapter 3** shows details of the database structure and the methodology for the recommendation model. Subsequently, the main functionalities of the application will be discussed in **Chapter 4**, and finally, **Chapter 5** is the conclusion and future work development directions.

## 1.3 The project implementation plan

A screenshot of a computer

Description automatically generated

A screenshot of a chat

Description automatically generated

*Figure 1: The project implementation plan*

# CHAPTER 2: THEORETICAL BASIS

## 2.1 .NET Framework

.NET Framework [[1](#bookmark=id.1nia2ey)] was first released in 2002. It is a software development platform constructed and nurtured by the “technological behemoth”, Microsoft. This platform enables us to build Windows applications, mobile apps, web applications, and primarily service-oriented platforms on the Windows operating system. It provides a virtual environment for executing programming codes, alongside a wide range of tools, libraries, and standards to construct and run applications. These encompass tasks such as memory management, software security, and exception handling.

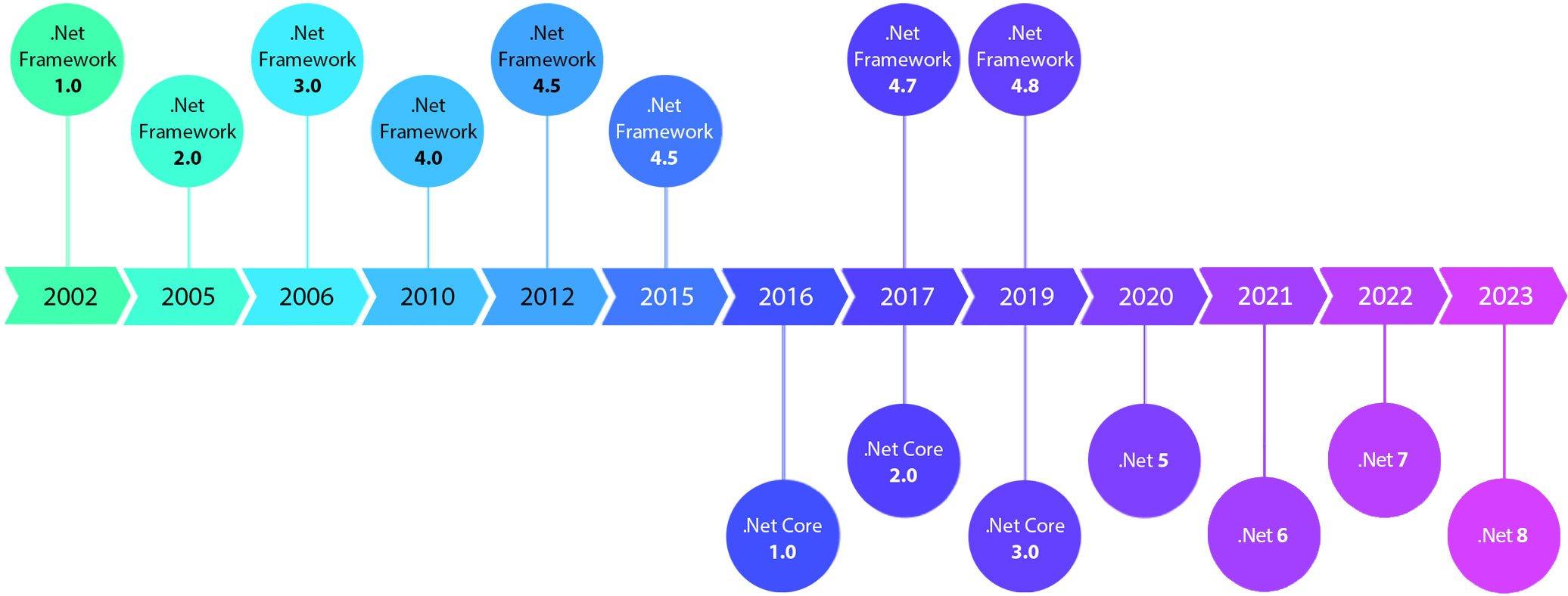


*Figure 2: .NET Framework (source: www.extranerds.com )*

The .NET Framework comprises several key features, including supporting developers in building applications using various programming languages such as C#, C++, F#, and more. It operates and executes source code within a software environment, distinct from the hardware environment, known as the Common Language Runtime (CLR). Furthermore, it assists in automatic memory management, enhancing application runtime speed while alleviating the burden on developers. Additionally, it facilitates software interconnection, as well as connecting software with databases and more.

Imagine yourself as a programmer wanting to create an application. Thanks to .NET Framework, you do not need to build the software entirely from scratch. Instead, you can combine pre-designed and built components of the application.

.NET Framework has undergone various versions, each with distinct features and improvements. The most significant milestone was .NET Framework 4.8. However, starting from .NET Core 3.0, Microsoft transitioned to .NET 5, followed by .NET 6. This shift has established a unified and more modern .NET ecosystem, catering to multiple platforms.



*Figure 3: History of .NET Framework (source: world.optimizely.com)*

### 2.1.1 Window Form

Windows Forms, often abbreviated as WinForms, serves as a graphical user interface (GUI) framework within the Microsoft .NET ecosystem. It encompasses a collection of classes and controls, enabling software developers to construct feature-rich and interactive desktop applications explicitly designed for the Microsoft Windows operating system.

Being among the initial GUI frameworks in the .NET landscape, Windows Forms is well-suited for constructing traditional desktop applications that offer a sense of familiarity in their appearance and functionality. Nevertheless, the emergence of more recent technologies like Windows Presentation Foundation (WPF) and Universal Windows Platform (UWP) has introduced an array of alternatives for application development.

Several notable attributes and components characterize Windows Forms:

* **Controls:** WinForms provides a diverse array of controls, encompassing buttons, text boxes, labels, list boxes, combo boxes, and more. These elements can be effortlessly incorporated into application windows to establish user interfaces.
* **Event Handling:** Developers possess the capability to attach event handlers to controls, facilitating responses to user actions such as button clicks, mouse movements, and keyboard inputs.
* **Layout Management:** WinForms offers tools for managing layouts, including panels, group boxes, and tab controls, which aid in organizing and positioning other controls within the application window.
* **Data Binding:** Windows Forms supports data binding, facilitating the automatic population and updating of controls with data derived from various sources such as databases.
* **Drag-and-Drop Design:** WinForms applications can be designed through a drag-and-drop interface within development environments like Visual Studio, simplifying the process of visually constructing user interfaces.
* **Customization:** Developers are empowered to generate custom controls and user interfaces tailored to the unique requirements of their applications.
* **Graphics and GDI+ Support:** Windows Forms enables the creation of graphics and incorporates GDI+ for rendering images, shapes, and other visual components.

### 2.1.2 LINQ

LINQ (Language Integrated Query) is a powerful feature in the .NET framework that enables developers to query and manipulate data from different data sources using a consistent and intuitive syntax. LINQ allows you to write queries directly within your programming language (such as C# or Visual Basic.NET) to retrieve, filter, transform, and manipulate data from various sources like databases, collections, XML, and more.

LINQ provides a unified query syntax, regardless of the type of data source, which makes it easier to work with data and reduces the need for complex loops and conditional statements. LINQ queries are strongly typed and checked by the compiler, which helps catch errors early in the development process.

LINQ supports various query operations, including filtering, sorting, grouping, joining, and aggregation. It is not limited to working with just collections but extends to databases (LINQ to SQL), XML (LINQ to XML), and more.

## 2.2 ML.NET Framework

ML.NET Framework is an open-source library by Microsoft [[2](#_heading=h.1idq7dh)] used for developing machine learning applications using the C# programming language and .NET platform. "ML" in ML.NET stands for "Machine Learning."



*Figure 4: ML.NET Framework (source: https://en.wikipedia.org/wiki/ML.NET)*

ML.NET provides a range of tools, libraries, and APIs to help developers easily integrate machine learning capabilities into their applications. This enables you to create and deploy machine learning models to perform tasks such as classification, prediction, clustering, and many other tasks that machine learning can assist with.

A strong point of ML.NET Framework is its seamless integration with existing .NET applications, sparing developers the need to learn a new language or go through complex steps to integrate machine learning into their applications.

ML.NET offers various tools and classes to help you preprocess data, build, train, and evaluate machine learning models. Additionally, it supports integrating trained models into real-world applications to directly perform predictions and classifications.

### 2.2.1 Automated Machine Learning

AutoML (Automated Machine Learning) is a powerful feature available in ML.NET, a machine learning framework developed by Microsoft. AutoML simplifies and accelerates the process of building machine learning models by automating several key steps, such as data preprocessing, feature engineering, algorithm selection, hyperparameter tuning, and model evaluation. It aims to make machine learning more accessible to developers and data scientists with varying levels of expertise. In the context of ML.NET, AutoML provides the following capabilities such as: Automated Model Selection, Hyperparameter Tuning, Feature Engineering, Model Evaluation and Selection,… AutoML in ML.NET makes it easy for developers to try out various machine learning methods and pick the best model, all without needing to be experts in machine learning. This speeds up model creation and lets users harness the benefits of machine learning more effectively. Using AutoML in ML.NET is a simple process:

* **Prepare Data:** Load and process your data using ML.NET's data tools.
* **Configure AutoML:** Set up an AutoML experiment with dataset details, target column, and desired model quality.
* **Train with AutoML:** Utilize the AutoML API to train your model. AutoML tests various algorithms and hyperparameters automatically.
* **Deploy the Model:** After training, deploy the best model to predict new data.

## 2.3 Recommendation system

A Recommendation System, also known as a Recommender System, is a subset of information filtering and data mining techniques used in various applications to provide personalized suggestions, recommendations, or predictions to users. The main goal of a recommendation system is to assist users in discovering items or content that they are likely to be interested in, based on their past behaviors, preferences, or similar patterns exhibited by other users.

Recommendation systems are widely utilized in a range of industries and contexts, including e-commerce, online streaming services, social media platforms, news websites, and more. They enhance user experiences by helping individuals discover relevant content or products, thereby increasing engagement and user satisfaction. There are some approaches to building recommendation systems.

* **Collaborative Filtering:** This approach makes recommendations by analyzing the historical behavior and preferences of users, identifying patterns and similarities between users or items. Collaborative filtering can be user-based (recommending items based on similar users' preferences) or item-based (recommending items similar to those the user has already interacted with).
* **Content-Based Filtering:** Content-based filtering focuses on the attributes of items or content and makes recommendations based on the similarity between the attributes of items and users' preferences. For example, recommending movies based on genres, actors, or directors.
* **Deep Learning:** Recent advances in deep learning techniques have been applied to recommendation systems, leveraging neural networks to capture complex patterns and relationships in user-item interactions.
* **Hybrid Methods:** Hybrid methods combine multiple recommendation techniques to improve accuracy and address limitations of individual methods.

## 2.4 Sentiment Classification

Sentiment analysis, a subset of natural language processing (NLP), is a compelling application that enables organizations to gauge the emotional tone behind textual data. ML.NET, Microsoft's open-source machine learning framework, empowers developers with the tools to perform sentiment classification effortlessly in C#, making it an invaluable asset for extracting actionable insights from textual content.

* **Understanding Sentiment Classification:** Sentiment classification involves determining the sentiment expressed in text – whether it is positive, negative, or neutral. By harnessing machine learning techniques, ML.NET enables the automatic categorization of text into sentiment classes, thus facilitating a deeper understanding of customer opinions, reviews, and feedback.
* **Key Components of Sentiment Classification with ML.NET:**
* **Data Preparation:** The foundation of any machine learning project, including sentiment classification, lies in data preparation. ML.NET provides a seamless way to preprocess and tokenize text data, converting it into a format that machine learning models can comprehend.
* **Feature Engineering:** Extracting relevant features from text is pivotal for accurate sentiment analysis. ML.NET offers various techniques, such as bag-of-words and TF-IDF (Term Frequency-Inverse Document Frequency), to transform text into numerical representations that machine learning algorithms can process.
* **Model Selection and Training:** ML.NET offers a spectrum of classification algorithms that can be utilized for sentiment analysis, such as logistic regression, support vector machines, and deep learning models. Developers can train these models using labeled datasets to learn the relationships between textual features and sentiment labels.
* **Evaluation and Tuning:** Rigorous evaluation of model performance is crucial. ML.NET provides tools to assess metrics like accuracy, precision, recall, and F1-score. Fine-tuning hyperparameters and selecting the best model variant can significantly enhance sentiment classification results.
* **Deployment and Inference:** Once trained, the sentiment classification model can be deployed within applications, allowing real-time sentiment analysis of incoming textual data. ML.NET's integration capabilities facilitate smooth model deployment to various platforms.

# CHAPTER 3: METHODOLOGY FOR A RECOMMENDATION MODEL

## 3.1 Database structure

We designed the booking travel database including Reviews, Customers, Partners, Destinations, Bookings and Admins like Figure 4 below.

A computer screen shot of a computer

Description automatically generated

*Figure 5: Database diagram*

## 3.1.1 Reviews table

*Table 1. Reviews table*

| **Column name** | **Datatype** | **Description** | **Key** | **Allow Nulls** |
| --- | --- | --- | --- | --- |
| ReviewID | int | The review of user about a location | PK | ☐ |
| DestinationID | int | The code of location | FK | ☐ |
| CustomerID | int | The code of users who book the location | FK | ☐ |
| Rating | int | Customer rating score from 1 to 5 |  | ☐ |
| ReviewText | text | The content of the review |  | ☑ |
| ReviewDate | date | The date when users rate and review |  | ☑ |
| Image | varchar(255) | A picture about location or the user experience |  | ☑ |

## 3.1.2 Partners table

*Table 2: Partners table*

| **Column name** | **Datatype** | **Description** | **Key** | **Allow Nulls** |
| --- | --- | --- | --- | --- |
| PartnerID | int | The code of partner who provided the detailed | PK | ☐ |
| PartnerName | varchar(100) | The name of partner |  | ☐ |
| ContactPerson | varchar(50) | Contact information for the representative of the partner |  | ☐ |
| Email | varchar(100) | Email address of the partner |  | ☐ |
| Phone | varchar(20) | Phone number of the partner |  | ☐ |

## 3.1.3 Destinations table

*Table 3: Destinations table*

| **Column name** | **Datatype** | **Description** | **Key** | **Allow Nulls** |
| --- | --- | --- | --- | --- |
| DestinationID | int | The code of location | PK | ☐ |
| DestinationName | varchar(255) | The name of location |  | ☐ |
| Location | varchar(100) | Geographical location |  | ☐ |
| Description | text | Detailed description of the location |  | ☑ |
| BasePrice | float | Original price of the tour that the customer must pay |  | ☐ |
| PriceForChildren | float | Price for children |  | ☑ |
| Discount | decimal(5,2) | Promotions that user can receive |  | ☑ |
| Images | varchar(255) | The picture of location |  | ☑ |
| PartnerID | int | The code of partner who provided the detailed location | FK | ☑ |
| TourismType | varchar(200) | The type of the tourism |  | ☑ |

## 3.1.4 Bookings table

*Table 4: Bookings table*

| **Column name** | **Datatype** | **Description** | **Key** | **Allow Nulls** |
| --- | --- | --- | --- | --- |
| BookingID | int | The booking code | PK | ☐ |
| DestinationID | int | The code of location | FK | ☐ |
| CustomerID | int | The code of users who book the location | FK | ☐ |
| PartnerID | int | The code of partner who provided the detailed location | FK | ☐ |
| BookingDate | date | The date of the booking that is booked by user |  | ☐ |
| BookingTime | time(7) | The time of the booking that is booked by user |  | ☐ |
| NumberOfAdults | int | The number of adults in the user’s booking |  | ☑ |
| NumberOfChildren | int | The number of children in the user’s booking |  | ☑ |
| TotalPrice | float | Total of the money that user should pay |  | ☑ |
| PaymentStatus | varchar(50) | The status of user’s payment |  | ☑ |

## 3.1.5 Customers Table

*Table 5: Customers Table*

| **Column name** | **Datatype** | **Description** | **Key** | **Allow Nulls** |
| --- | --- | --- | --- | --- |
| CustomerID | int | Code of users | PK | ☐ |
| FirstName | varchar(50) | The first name of user |  | ☐ |
| LastName | varchar(50) | The last name of user |  | ☐ |
| Email | varchar(100) | Email address of user |  | ☐ |
| Phone | varchar(20) | Phone number of user |  | ☐ |
| Password | varchar(100) | User login password |  | ☐ |

## 3.1.6 Admins table

*Table 6: Admins table*

| **Column name** | **Datatype** | **Description** | **Key** | **Allow Nulls** |
| --- | --- | --- | --- | --- |
| AdminID | int | Unique key for Admin | PK | ☐ |
| Username | varchar(50) | Admin login name |  | ☐ |
| Password | varchar(50) | Admin login password |  | ☐ |

## 3.2 Recommendation model



*Figure 6: Diagram for experiment progress of Recommendation system.*

* **Sentiment Analysis:** The sentiment analysis model is built using ML.NET. It takes tourist comments as input and predicts the sentiment score. The sentiment score represents the positivity or negativity of the comment. The model has been trained on a labeled dataset, and it performs well in analyzing comments and assigning sentiment scores. Positive and negative sentiment scores are calculated, and the percentage of positive sentiment is computed based on the total sentiment score. This percentage will be used to adjust the user rating.
* **Recommendation System:** The recommendation system utilizes collaborative filtering to suggest travel destinations to users. The main components are:
* **Data Collection and Preparation:** User ratings, sentiment scores, and destination data are collected and processed. This data is used to create a user-item interaction matrix.
* **Model Building:** Collaborative filtering models are trained using the user-item interaction matrix. The model takes user and item IDs along with sentiment-adjusted ratings as input and predicts a rounded score for each destination.
* **Recommendation Generation:** For each user, the model generates a list of recommended destinations based on the predicted rounded scores. Destinations with higher scores are recommended to users.
* **Expected Process:** Tourists use the travel booking application to book destinations and provide comments.
* **Sentiment Analysis:** Tourist comments are analyzed using the sentiment analysis model. The sentiment scores and percentages are computed.
* **Rating Adjustment:** User ratings are adjusted based on the sentiment percentage to account for positive or negative feedback.
* **Recommendation Generation:** The recommendation system generates travel destination recommendations for users using collaborative filtering.
* **Display Recommendations:** Users can view the recommended destinations in the application's user interface.

Here is how we implemented and got the result:

The result is not good. The challenges in successfully implementing the machine learning model were primarily attributed to the utilization of synthetic data stored in the SQL database. This dataset lacked authenticity and realism, and its size was notably limited. Additionally, the team faced constraints in terms of their proficiency in machine learning techniques.

Consequently, the algorithm's recommendation outcomes were notably suboptimal, resulting in inadequately tailored suggestions to users. In light of these limitations, the team has proposed potential enhancements. They are considering the acquisition of genuine and expansive data from reputable sources such as Kaggle. This approach seeks to introduce more representative and diverse data to bolster the recommendation system.

Furthermore, the team has acknowledged the feasibility of generating additional data to enrich the existing dataset. A pivotal consideration is the transition from automated model development to a customized approach. By tailoring the model to the specific nuances of the travel booking domain, the team aims to achieve improved recommendation accuracy.

This strategic direction demonstrates the team's commitment to leveraging external resources for enriched data, skillful customization of machine learning models, and a forward-looking approach towards the integration of a refined recommendation system that aligns with user preferences.

# 

# CHAPTER 4: EXPERIMENTAL APPLICATION

A diagram of a system

Description automatically generated

*Figure 7: Use case of Travel booking application*

Based on the Use Case diagram, the model briefly describes four main functions of the travel booking system with specific functions serving only for each of the four Actors of the system. The four main actors of the system are admin, user, partner, and system, and their detailed functions were described as follows:

* **Admin:** They are responsible for managing the system including:
* **Managing user information:** Admin can deactivate users who violate the organization’s policies and laws, such as using bots, spreading inappropriate content, provoking others, etc. Admin can also CRUD other admin accounts to operate the system.
* **Managing partner information about destinations:** In this function, Admin can CRUD destinations provided by various partners. Admin can also CRUD the parties that cooperate with the company on the interface based on their contributions.
* **Managing transactions:** The company does not directly process payments but uses another intermediary platform, but the system still receives and updates user information to update the status of their orders on the application.
* ***User*:** All users who download and operate with user accounts through the travel booking application can view options for destinations and related services. They can make changes related to their personal profile and book and complete payments. Most importantly, customers can CRUD reviews related to the places they have been to, meaning they can only comment when they have completed the journey they have booked.
* ***Partner*:** Includes both partners providing travel amenities related to destinations and third parties completing customer payments for orders. All will be notified by the system if an event occurs
* ***System*:** The intermediary that facilitates the exchange of users and the unit responsible for providing services. This is also where the recommendation model based on user reviews about destinations is equipped and will make appropriate suggestions to users based on the results of the model obtained.

## 4.1 Function: Login

The login function is a critical component and a compulsory task of any application, including a traveling booking application that manages account access, and secures and protects users’ privacy of accessors' actions on the built application. The system should use strong encryption to protect users' passwords, and it should only collect the necessary information from users. The login function should also be easy to use and understand. Users should be able to log in quickly and easily, and they should be able to understand the information that is being collected from them. By the way, this function also specifies users' and admins’ needs on application execution.

### 4.1.1 DFD

A diagram of a login

Description automatically generated

*Figure 8: Data Flow Diagram Level 0 for the Login Function*

As mentioned, the system will record 2 login account objects, which are administrators and users, while partners will be additional administrators based on the company’s requirements. the login function for Admin and users in the travel booking application typically includes the following steps:

* The user or admin enters their email address and password.
* The system verifies the user's credentials.
* If the user's credentials are valid, they are logged in.
* If the user's credentials are invalid, they are prompted to enter their credentials again.

Overall, there are two needed dimensions for storing account information, which are the user database for admin and users and the partner database to update changes executed by Admins. By the way, our application will provide different execution windows for two main account types.

A diagram of a user account

Description automatically generated

*Figure 9: Data Flow Diagram Level 1 for the Login Function*

In DFD Level 1, each activity that can be performed in the Login function is clearly represented: For Super admins, they can initialize login accounts and assign operational roles for another admin; the admin who manages the application can edit the activity list of users and partners after completing role authentication through login accounts. Both users and admins can adjust their personal information, but admins will be limited to editing some login account information fields, which help the company manage admins. At the same time, the system also allows the creation of a knowledge base to support navigation activities suitable for each application user. To protect user information on the application, the system will automatically exit and log out when no activity is recorded within 45 minutes. By the way, users can customize to turn on or off this function if not necessary.

### 4.1.2 BPMN

A diagram of a flowchart

Description automatically generated

*Figure 10: BPMN for Login function*

The BPMN graph has described the process that users and admin will perform to log in to the system, in which the diagram also refers to the system identifying the visitor’s role based on the login account.

### 4.1.3 Mock-ups

A person taking a picture of herself

Description automatically generated

*Figure 11: The welcome screen*

## 4.2 Function: Booking

### 4.2.1 DFD

**A diagram of a travel application

Description automatically generated**

*Figure 12: DFD level 0 for the Booking function*

Similar to Login, there are two separate active screens provided for either admin or user accounts. The booking function for users in a recommendation based on users' reviews in a travel booking application typically includes the following steps:

* The user selects a destination and date range.
* The system generates a list of recommended travel arrangements based on the user's past reviews and the reviews of other users.
* The user selects a travel arrangement from the list.
* Since users selected “Continue booking by completing payment”, the system will automatically switch to partners’ site to handle the transaction.
* After completing the transaction, partners will send status to update for booking database.
* The system confirms the booking and sends a confirmation email to the user.

The booking function should be secure and protect users' financial information. The system should use strong encryption to protect users' payment information, and it should only collect the necessary information from users. The booking function should also be easy to use and understand. Users should be able to book their travel arrangements quickly and easily, and they should be able to understand the information that is being collected from them.

Here are some additional considerations for the booking function in a recommendation based on users' reviews in a travel booking application:

* The booking function should be available on all devices, including mobile phones and tablets.
* The booking function should be accessible to users with disabilities.
* The booking function should be secure and protect users' financial information.
* The booking function should be easy to use and understand.

By following these considerations, developers can create a booking function that is secure, user-friendly, and accessible to all users.

In addition to the above, here are some specific considerations for the booking function for Admin and users in a recommendation based on users' reviews in a travel booking application:

* For Admin, the booking function should allow them to view all bookings that have been made through the system. This will allow Admin to track the status of bookings and to resolve any issues that may arise.
* For users, the booking function should allow them to view their past bookings and modify their bookings as needed. However, following the rules of each case, users may be restricted from editing or cancelling bookings. This will give users peace of mind, knowing that they can easily make changes to their travel arrangements if necessary.

By taking into account the specific needs of Admin and users, developers can create a booking function that is both efficient and user-friendly.

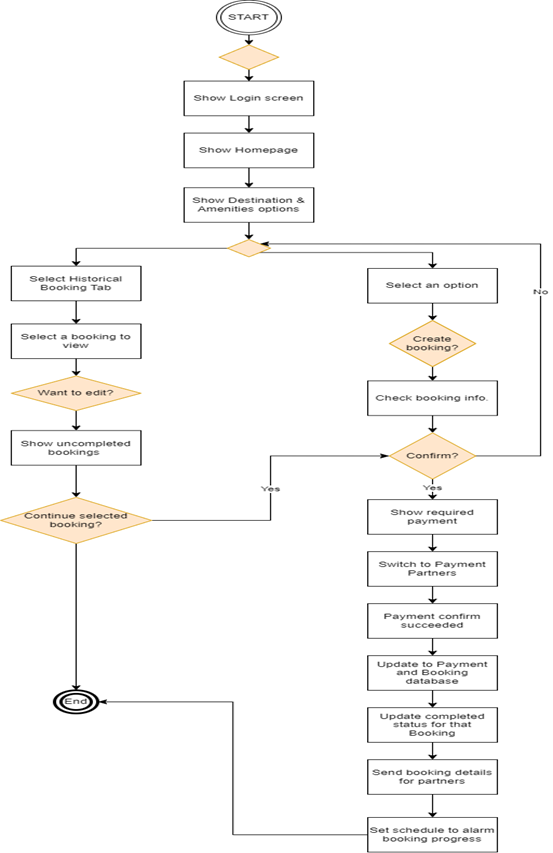
A diagram of a system

Description automatically generated

*Figure 13: DFD level 1 for the Booking function*

There are 4 main databases that should be generated for this function, which are User, Payment, Booking and Destination database.

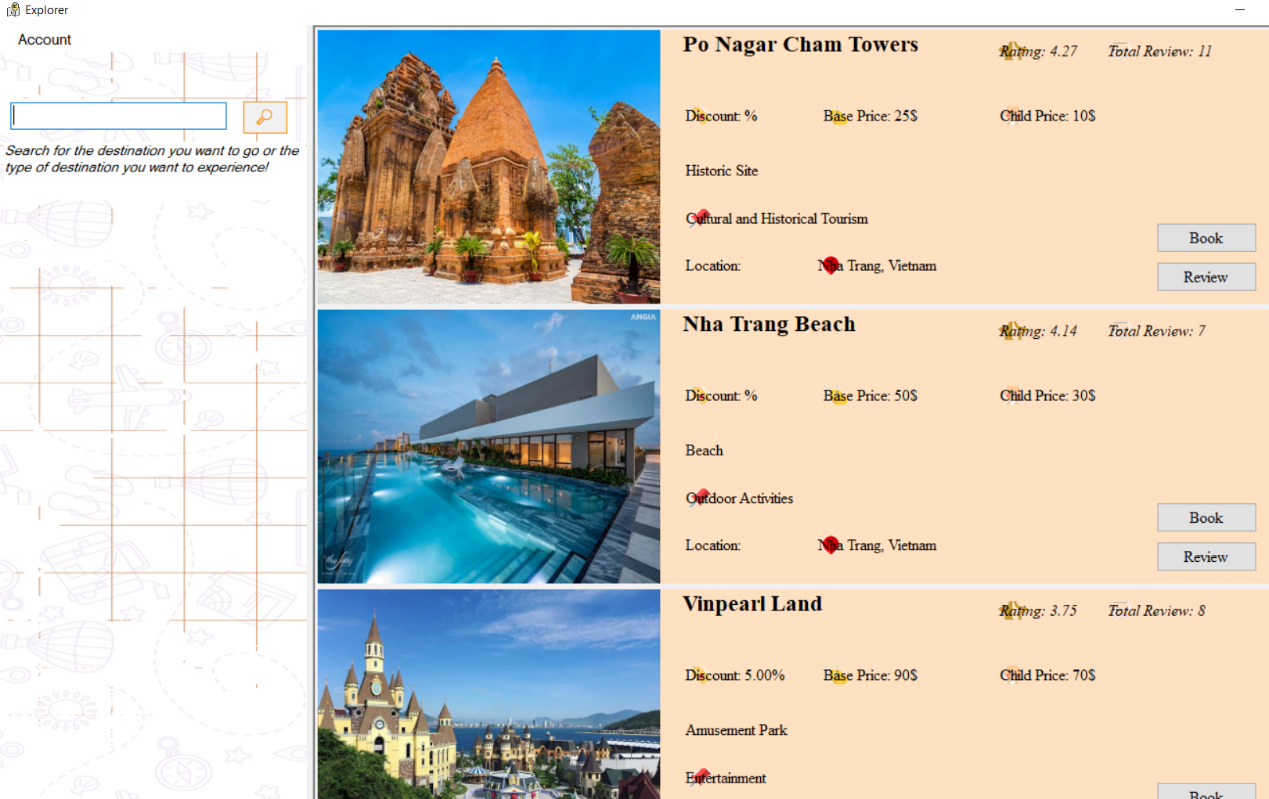
### 4.2.2 BPMN



*Figure 14: BPMN for Booking function*

According to our business rules, users’ booking labeled uncompleted will be updated to knowledge base of the application and remind or request users to complete later.

### 4.2.3 Mock-ups

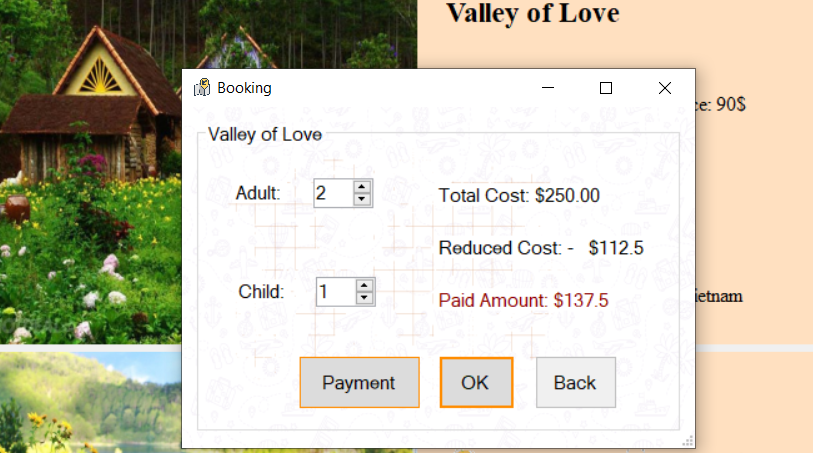


*Figure 15: : The explorer screen when the user choose Explore button*

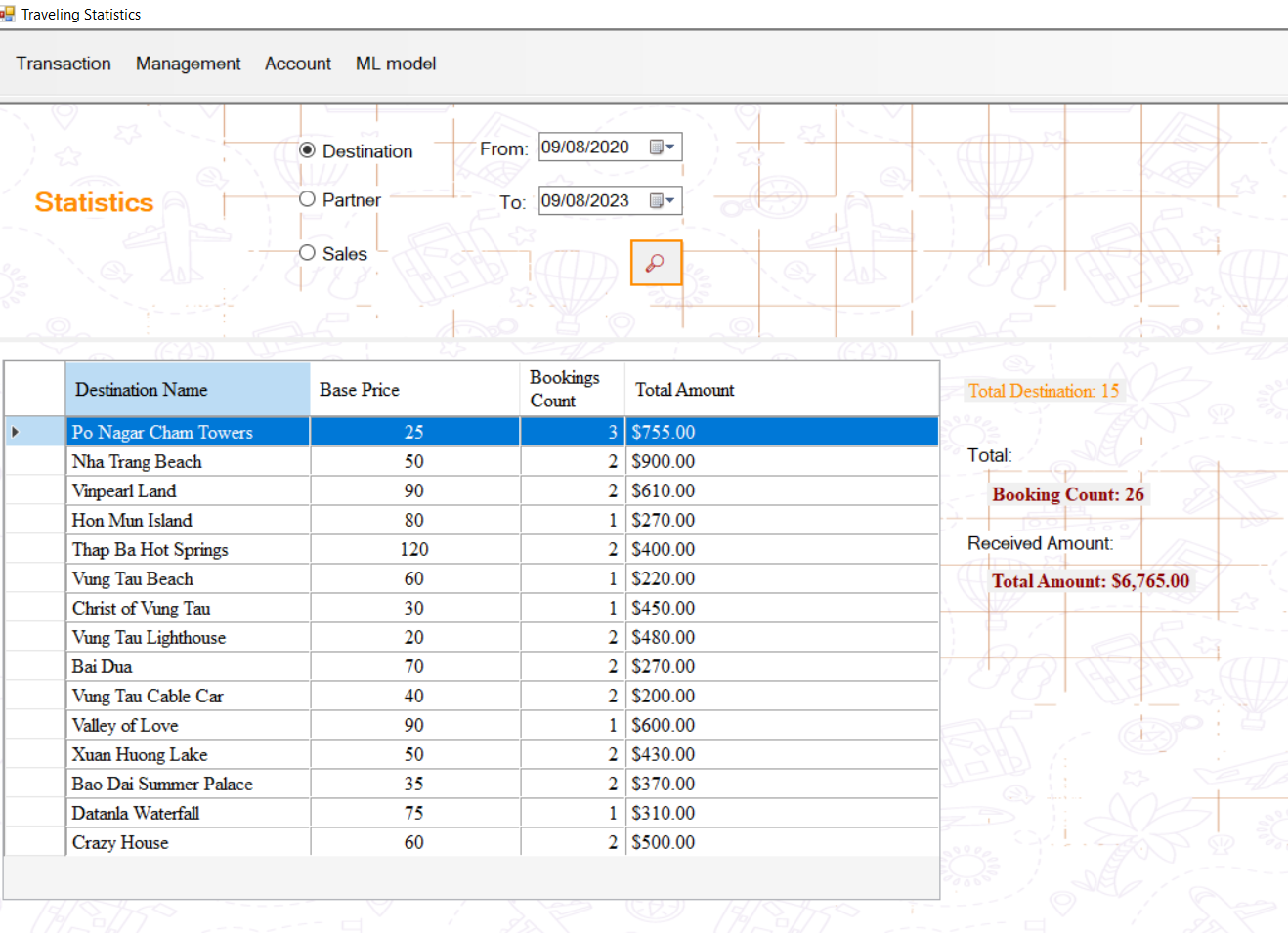
A screenshot of a login screen

Description automatically generated

*Figure 16: The login screen*



*Figure 17: : The booking screen*



*Figure 18: The travel statistic screen*

A screenshot of a computer

Description automatically generated

*Figure 19: The manageable partner screen*

A screenshot of a computer

Description automatically generated

*Figure 20: The recommendation screen*

## 4.3 Function: Reviews management

### 4.3.1 DFD

A diagram of a travel application

Description automatically generated

*Figure 21: DFD level 0 for the Reviews management function*

The reviews function for users in a recommendation based on users' reviews in a travel booking application typically includes the following steps:

* The user selects a travel destination or service that they have recently visited.
* The user provides their feedback on the travel destination or service, including a rating and a written review.
* The system validates the user's feedback and stores it in the database.
* The system uses the user's feedback to generate recommendations for other users.

The reviews function should be easy to use and understand. Users should be able to provide their feedback quickly and easily, and they should be able to understand how their feedback is being used to generate recommendations. The reviews' function should also be secure and protect users' privacy. The system should only collect the necessary information from users, and it should use strong encryption to protect users' feedback.

Here are some additional considerations for the reviews function in a recommendation based on users' reviews in a travel booking application:

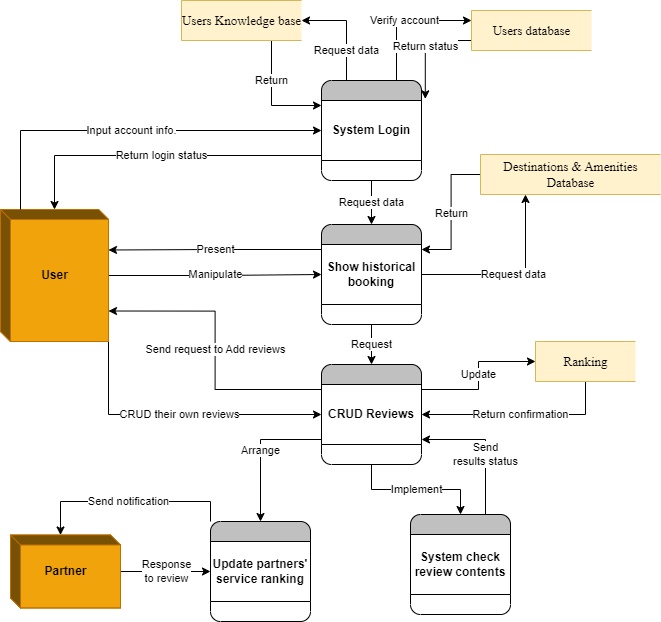
* The reviews function should be available on all devices, including mobile phones and tablets.
* The reviews function should be accessible to users with disabilities.
* The reviews function should be secure and protect users' privacy.
* The reviews function should be easy to use and understand.

By following these considerations, developers can create a reviews function that is secure, user-friendly, and accessible to all users.

Here are some specific examples of how the reviews function can be used by Admin in a recommendation based on users' reviews in a travel booking application:

* Admin can use the reviews function to monitor user feedback and identify any potential problems with travel destinations or services.
* Admin can use the reviews function to identify popular travel destinations and services that are likely to be of interest to other users.
* Users can use the reviews function to get feedback from other users on travel destinations and services that they are considering visiting.
* Users can use the reviews function to share their own feedback on travel destinations and services with other users.

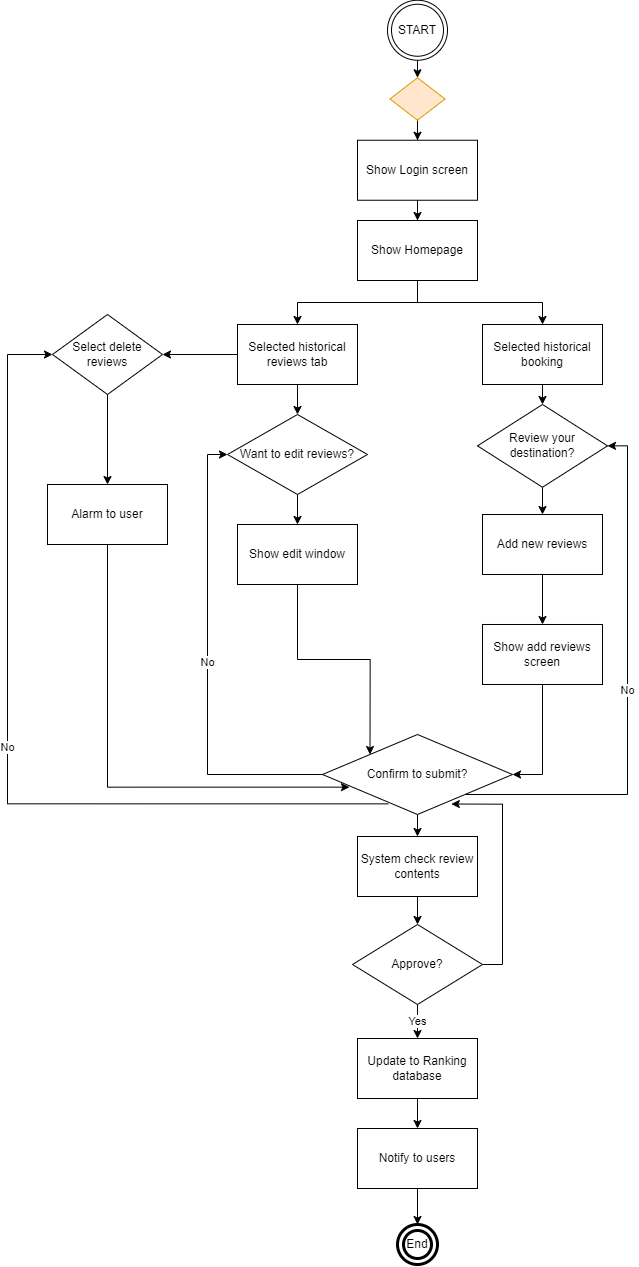
By providing a comprehensive and user-friendly reviews function, travel booking applications can help users make informed decisions about where to travel and what services to use.



*Figure 22: DFD level 1 for the Review management function*

Futhuremore, in the DFD level 1 has mentioned the reviews from users will become criteria to ranking partners.

### 4.3.2 BPMN



*Figure 23: BPMN for the Reviews management function*

In our case, users allow to generate reviews for their own booking. Our system will automatically encourage our users to generate reviews via notification and ranking to users’ contribution.

### 4.3.3 Mock-ups

A screenshot of a computer

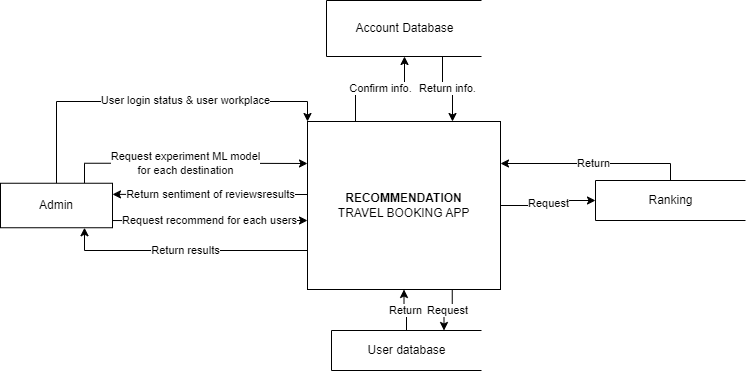
Description automatically generated

*Figure 24: The rating screen*

## 4.4 Function: Destination recommendations

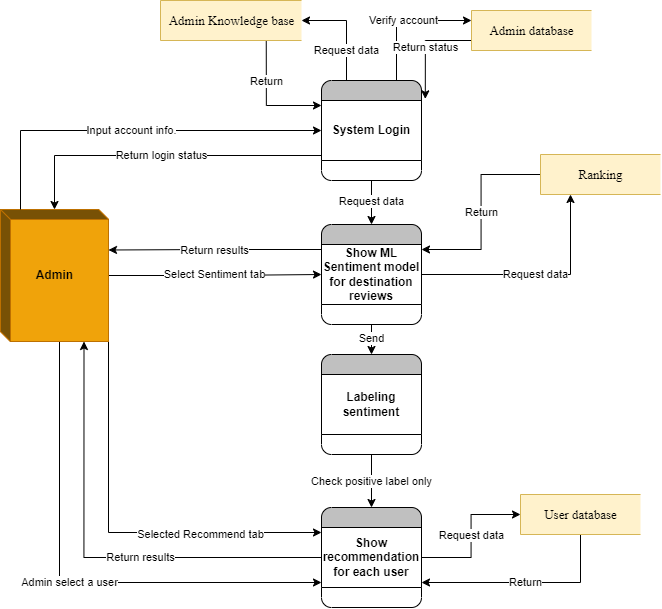
The main objective for this research is the Destination recommendation built from ML. Net Frameworks provided by Microsoft. Thanks to this, we have shortened our time to build a machine learning related to reviews sentiment to apply for recommendation system.

### 4.4.1 DFD



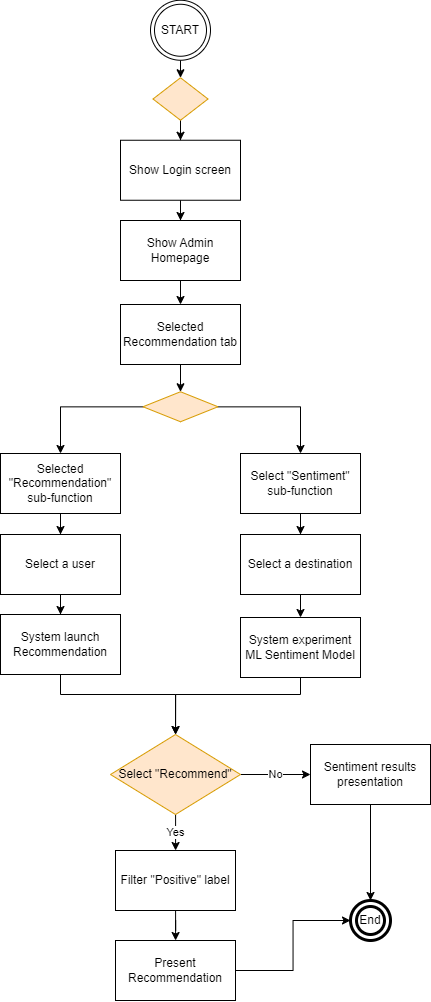
*Figure 25: DFD level 0 for Destination recommendation function*

In general, our application presented the results for sentiment or recommendation function only in Admin active screen. For users, these model will be integrated with showing destination options for users.



*Figure 26: DFD level 1 for Destination recommendation function*

### 4.4.2 BPMN



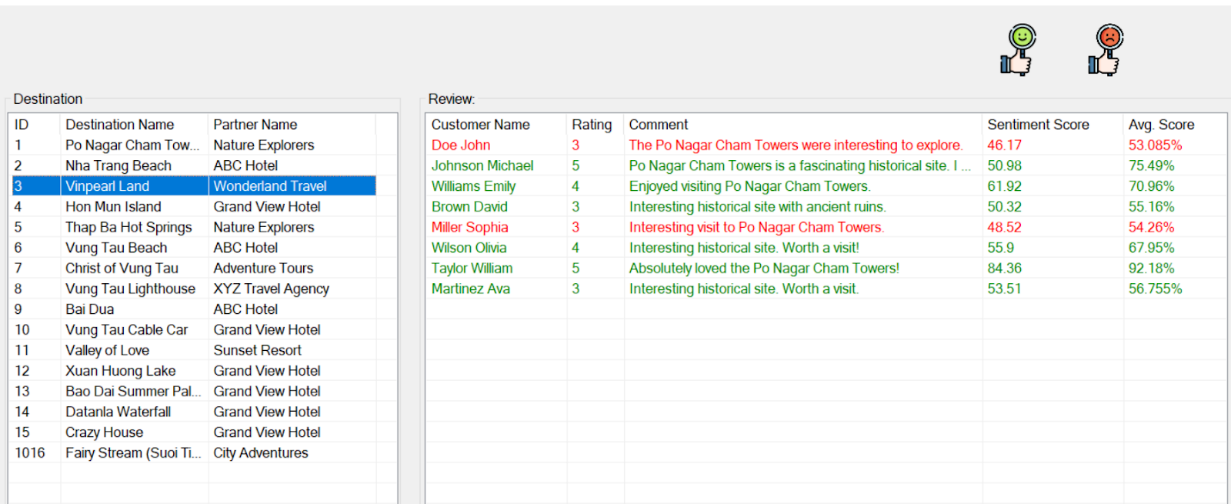
*Figure 27: BPMN for Destination recommendation function*

For recommendation system, they will use the results of review sentiment model as their output, where was labeled positive to recommend to users.

### 4.4.3 Evaluation model

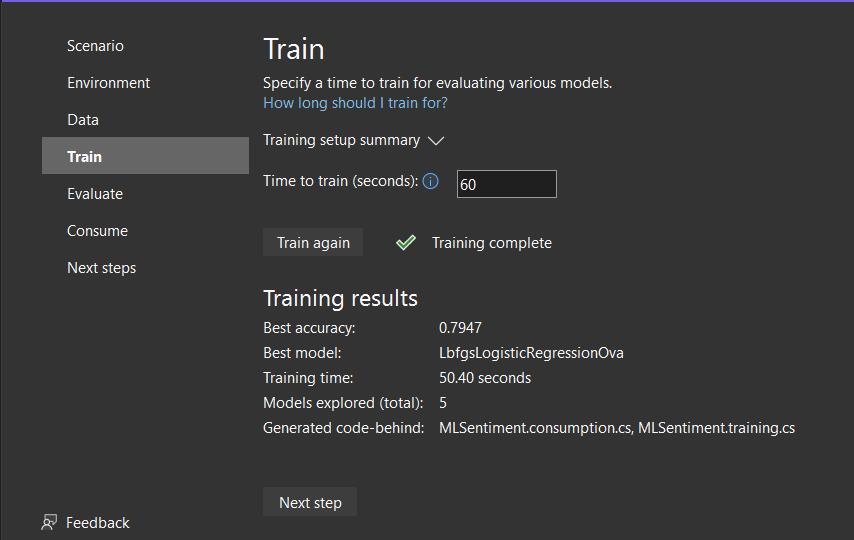
* **Sentiment Analysis**

The sentiment analysis model is built using ML.NET. It takes tourist comments as input and predicts the sentiment score. The sentiment score represents the positivity or negativity of the comment. The model has been trained on a labeled dataset, and it performs well in analyzing comments and assigning sentiment scores. Positive and negative sentiment scores are calculated, and the percentage of positive sentiment is computed based on the total sentiment score. This percentage will be used to adjust the user rating.



*Figure 28: The sentiment of the users about location after using ML model*

And the Sentiment model we trained have the result quite fine, as the data we trained, including 1000 records and of **Yelp Reviews Dataset,** but when we applied on our data reviews, we see our sentiment on some review not correct, as this is the valuation for our model.



*Figure 29: The result of model after training*

* **Recommendation System**

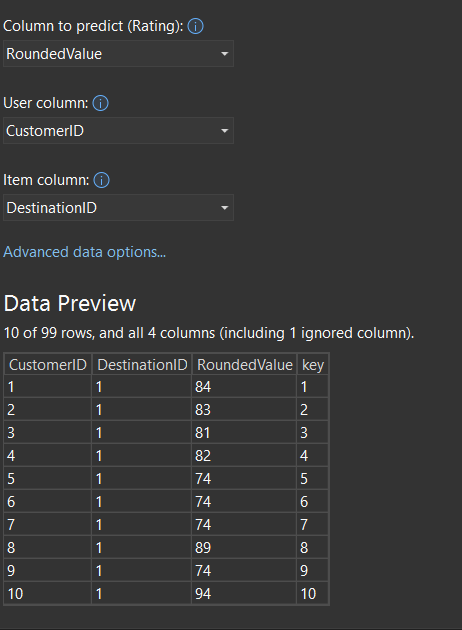
The recommendation system utilizes collaborative filtering to suggest travel destinations to users. The main components are:

* **Data Collection and Preparation:** User ratings, sentiment scores, and destination data are collected and processed. This data is used to create a user-item interaction matrix.
* **Model Building:** Collaborative filtering models are trained using the user-item interaction matrix. The model takes user and item IDs along with sentiment-adjusted ratings as input and predicts a rounded score for each destination.
* **Recommendation Generation:** For each user, the model generates a list of recommended destinations based on the predicted rounded scores. Destinations with higher scores are recommended to users.
* **Expected Process:**

Tourists use the travel booking application to book destinations and provide comments.

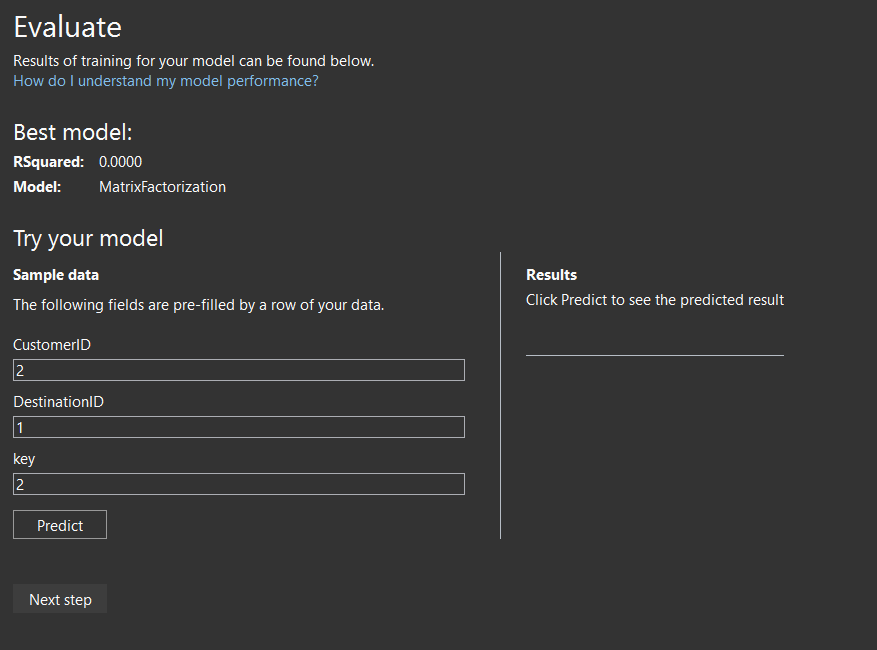
* **Sentiment Analysis:** Tourist comments are analyzed using the sentiment analysis model. The sentiment scores and percentages are computed.
* **Rating Adjustment:** User ratings are adjusted based on the sentiment percentage to account for positive or negative feedback.
* **Recommendation Generation:** The recommendation system generates travel destination recommendations for users using collaborative filtering.
* **Display Recommendations:** Users can view the recommended destinations in the application's user interface.

Here is how we implemented and got the result:



*Figure 30: The configure information of model*

And the figure below is the evaluation that the model got:



*Figure 31: The evaluation of the recommendation*

And the result is not good, The challenges in successfully implementing the machine learning model were primarily attributed to the utilization of synthetic data stored in the SQL database. This dataset lacked authenticity and realism, and its size was notably limited. Additionally, the team faced constraints in terms of their proficiency in machine learning techniques.

Consequently, the algorithm's recommendation outcomes were notably suboptimal, resulting in inadequately tailored suggestions to users. In light of these limitations, the team has proposed potential enhancements. They are considering the acquisition of genuine and expansive data from reputable sources such as Kaggle. This approach seeks to introduce more representative and diverse data to bolster the recommendation system.

Furthermore, the team has acknowledged the feasibility of generating additional data to enrich the existing dataset. A pivotal consideration is the transition from automated model development to a customized approach. By tailoring the model to the specific nuances of the travel booking domain, the team aims to achieve improved recommendation accuracy.

This strategic direction demonstrates the team's commitment to leveraging external resources for enriched data, skillful customization of machine learning models, and a forward-looking approach towards the integration of a refined recommendation system that aligns with user preferences.

# CHAPTER 5: CONCLUSIONS AND FUTURE WORKS

## 5.1 Results

After working diligently, our team has achieved a significant success with our travel booking app. We've added a smart feature that can analyze and understand customer feedback, helping businesses know how customers feel about their experiences.

This achievement demonstrates our dedication to using technology to make better decisions. With our new feature, businesses can discover important trends and patterns in what customers say. This helps them make smarter choices, whether it's improving customer service, making ads more effective, or enhancing their products.

While we faced some challenges with our recommendation feature due to limited data and skills, we're excited about the future. As we learn more and get better data, we can make our recommendation system stronger. This means our app could someday help travelers find the perfect destination based on their interests.

In summary, our successful customer feedback feature is a big step forward. It shows what our team can do and sets us up for even more exciting things in the future. We're committed to using technology to make our app even better and provide the best experiences for users."

## 5.2. Limitations

Despite the successful development and deployment of the application, there are still certain limitations. Due to time constraints, the application has not been fully optimized or diversified, and its functionalities remain at a basic level. Furthermore, the application has been built solely as a WPF (Windows Presentation Foundation) application, resulting in an interface that is not aesthetically appealing and lacks full optimization for user experience. Additionally, there are undoubtedly unforeseen errors within the application. We are committed to optimizing and upgrading the application in the future to address these issues.

## 5.3. Future works

In the future, our focus is directed towards developing the application on a web-based platform to fully leverage the abundant resources and modern technologies available today. This transition aims to enhance user experience by taking advantage of contemporary advancements, while also enabling broader accessibility to a wider range of user demographics. Additionally, it is imperative for us to continuously innovate and expand the application's capabilities to cater to a diverse set of user needs. Through these endeavors, we aspire to create a more versatile and feature-rich application that truly resonates with our users' requirements.

# REFERENCES

[1] <https://learn.microsoft.com/en-us/dotnet/framework/get-started/overview>

[2] <https://learn.microsoft.com/en-us/dotnet/machine-learning/how-does-mldotnet-work>

# APPENDIX

1. [The link of the detailed pictures in this report.](https://drive.google.com/drive/folders/1XSXBj0lDKxCW34kfETENjkH6gvjWEcy-?fbclid=IwAR1yNFDhb1wQl41rqaePkUWPtlFxRTQZA_ysY2V6OB9iLJpL5JgXQTz5QZ4)