



Universiteit Leiden

ICT in Business and the Public Sector

**Providing Insights into Software Usage through Data
Mining: Use Case Based User Profiling**

Name: Yi Zhang

Student-no: s2420597

Date: 1/1/2022

1st supervisor: Guus Ramackers

2nd supervisor: Peter van der Putten

Abstract

Understanding the actual usage of software products by end-users is vital for software developers and product managers in order to incrementally improve the software and tailor it to users' needs. Software usage data mining provides insights into the actual usage patterns of the software after it is deployed. Use case based user profiling is a method to divide the user base into different user clusters, each focusing on specific subsets of use cases of the overall product functionality. It applies data mining clustering techniques to group users based on their actual software usage data, allowing future software releases to be tailored to the needs of the user group segments.

UML use case models are used as a reference framework for the preparation, collection, and interpretation of usage data. The results of this analysis are presented in the form of an annotated version of use case diagrams, called "infographic use case diagrams." These illustrate the actual interaction between end-users and the system by different user groups. Since use case models are the primary method for capturing software requirements prior to implementation, our approach provides an effective feedback mechanism between the initial design and the actual use of the software. In this way, we support the development of software products based on insights from design and use during runtime.

In our research, we analyse the efficiency and effectiveness of various clustering techniques in this context. In particular, we review the performance of the K-means clustering algorithm and the hierarchical clustering algorithm in grouping users into unique user types. In addition, we propose a design for an automated tool solution for our Software Usage Data Mining method.

Acknowledgement

I want to thank my first supervisor Dr Guus Ramackers for supporting me in completing the research project. We come up with this innovative design together. Furthermore, Dr Ramackers gave me many suggestions in choosing appropriate software to develop the software prototype. Also, many thanks to Dr Ramackers for giving me sufficient time to finish up my master's research and bring the idea of use case based user profiling into life.

Thank Dr Peter van der Putten for being my 2nd supervisor. He gave me precious suggestions in developing the methodology and enriching thesis contents.

Contents

1	Introduction.....	7
1.1	Improve Project Management with User Segmentation and Usage Data.....	8
1.2	Improve Software Functionality.....	9
1.3	Discover a New Method of Creating User Profiles	9
1.4	Research Questions	9
1.5	Research Overview.....	10
2	Background and Related Work	12
2.1	Use Case Models	12
2.2	Software Usage Data Mining.....	13
2.3	User profiling	15
2.4	Cluster Analysis Techniques.....	16
2.4.1	K-means Clustering Algorithm and Hierarchical Clustering Algorithm Introduction	17
2.4.2	Data Normalization and Standardization.....	18
2.5	Chapter Summary	19
3	Use Case Based User Profiling Process.....	20
3.1	Use Case Based User Profiling Process Design	20
3.1.1	Business Understanding with Use Case Models.....	21
3.1.2	Data Understanding and Preparation.....	22
3.1.3	Cluster Analysis.....	22
3.1.4	Plotting and Evaluation.....	23
3.2	Chapter Summary	23
4	Use Case Based User Profiling in a Help Desk Interaction System.....	24
4.1.1	Help Desk Interaction System Business Understanding	24
4.1.2	Data Preparation.....	26
4.1.3	K-means Cluster Analysis without Data Normalization.....	27

4.1.4	K-means Cluster Analysis with Data Normalization	31
4.2	Chapter Summary	33
4.3	Supplement: A Coloured Infographic Use Case Diagram Design	34
5	Use Case Based User Profiling in a Web Blog Application	35
5.1	Software Components in the Experiment.....	35
5.2	Web Blog Application – A Blog Management System	38
5.3	User Input Simulation Method.....	42
5.3.1	User Input Simulation Objective.....	42
5.3.2	User Simulation for the Duration Domain	43
5.3.3	User Simulation for the Frequency Domain	44
5.3.4	Data Recording.....	44
5.3.5	Usage Data Dataset Overview.....	45
5.4	Data Aggregation Method.....	47
5.5	Use Case Based User Cluster Analyses Results.....	50
5.5.1	Duration Domain - K-means Clustering	50
5.5.2	Duration Domain - Hierarchical Clustering	55
5.5.3	Frequency Domain - K-means Clustering	59
5.5.4	Frequency Domain - Hierarchical Clustering.....	64
5.5.5	Frequency Domain - Hierarchical Clustering for User Subtypes.....	68
5.6	Result Comparison in the Duration Domain.....	72
5.6.1	Clusters Overview	72
5.6.2	User Type Distribution	74
5.6.3	User Type Comparison by Use Cases	75
5.6.4	Infographic Use Case Diagrams Comparison	76
5.6.5	Conclusions and Discussions in Duration Domain.....	77
5.7	Result Comparison in the Frequency Domain.....	78

5.7.1	Clusters Overview	78
5.7.2	User Type Distribution	80
5.7.3	User Type Comparison by Use Cases	81
5.7.4	Infographic Use Case Diagrams Comparison	81
5.7.5	Conclusions in Frequency Domain	83
5.8	Subtype Discovery.....	83
5.8.1	User Type Distribution	84
5.8.2	User Type Comparison by Use Cases	84
5.8.3	Infographic Use Case Diagram Showcase.....	85
5.8.4	Subtype Discovering Process Summary.....	87
5.9	Clustering Algorithm Choice Discussion	87
5.10	Chapter Summary	88
6	Integrated Tool Support for Use Case Based User Profiling	89
6.1	Interactive Features.....	89
6.2	The Feature of Improving Software Use Case Models Automatically	92
6.3	Summary	93
7	Conclusions	95
8	Reference	97
9	List of Tables.....	100
10	List of Figures	102
11	Appendix.....	104

1 Introduction

People share their experiences with software products in the form of written feedback. Developers collect this feedback to identify bugs, vulnerabilities, and new features in their software products. However, text-based feedback is difficult to manage. It takes time to understand and process this feedback (Morales-Ramirez et al., 2015). Moreover, the quality of this feedback cannot be guaranteed. It can be expressed in only a few words and can be ambiguous (Guzman et al., 2014). Developers need to know how people use their software in detail. They need much more detailed insight into the actual usage patterns of the software. They also need to understand the different types of software users (e.g., casual users, novice users, experienced users, and "power users").

Capturing user behaviour from usage data in real-time is a more effective strategy for learning user behaviours (Alexander et al., 2008). Deploying features to capture operational data automatically takes less time than capturing written feedback by a small proportion of users. For example, E-commerce companies use data mining tools to learn from the shopping habits of the different user segments. Data mining tools can determine which products are the most appealing to customers (Jiang & Yu, 2008). Web multimedia systems like YouTube examine users' viewing habits with usage data mining (Su & Wu, 2021).

A popular topic in the field of business intelligence is user classification with artificial intelligence technologies. Syadzali et al. (2020) applied supervised machine learning techniques to classify users in an online crowdfunding system. The classification system analysed user behaviour patterns that led users to donate to the projects. Similarly, companies can apply artificial intelligence in contact centres to analyse customer satisfaction (Godbole & Roy, 2008). Text classification can distinguish customers' arguments with products and their satisfaction with the service. Social media companies rely on learning user behaviours to target users' promotions and guide them to their interested topics or posts. In user classification in social media, artificial intelligence algorithms can categorize users into different user groups based on their characteristics and activities (Cheng et al., 2010). Social network companies like Twitter create user profiles that describe users' characteristics

(Pennacchiotti & Popescu, 2011) for user classification purposes. These user profiles include users' demographic information like gender, age, name, location, etc., the frequency and the number of users' activities on the social media system, the linguistic content held in the users' tweets, and users' social network structure information. Hence social media companies can target users precisely and even predict users' demographic attributes from their post content (Guimaraes et al., 2017).

Our research aims to develop a method that leverages the value of use case models in software user profiling. Our work proves that use case models can be used effectively in user behaviour pattern discovery. Use case models explain the process of how users interact with different system features (Qazi et al., 2016). They capture user requirements and are widely used in the design and development of software systems. Our use case based user profiling method can characterize users with software use case models and group these users into various user groups with similar behaviour patterns. As such, our user profiling method differs from the approaches described above because it provides a direct feedback loop between the models used in the initial design of a system and its subsequent usage by user segments after deployment. Consequently, software developers can quickly analyse software usage conditions and continuously improve their software products by modifying and extending the system design. This paper has three main research objectives, described in the following sections.

1.1 Improve Project Management with User Segmentation and Usage Data

Many software development teams use Scrum and Agile methods in software development, where developers rely heavily on user feedback to make continuous improvements (Schwaber & Beedle, 2002). Scrum and agile can deliver improvements in each iteration and respond to changing requirements. In this context, the development team should capture user requirements precisely. A clear understanding of how people use the software is therefore critical.

Gathering user feedback and usage data should also be flexible and fast. Investing too much time and effort in analysing feedback and using data can delay delivery. In addition, if the team does not receive correct information from the feedback, the quality of the publication

would be affected (Krusche et al., 2014). Therefore, we should reduce the time required to analyse tersely written feedback of a sub-section of users by using automatic mechanisms to collect effective feedback based on actual software usage. Furthermore, using an efficient automated analysis tool can improve the reliability of the generated results.

1.2 Improve Software Functionality

Another benefit that our approach offers software developers and companies is the incremental improvement of software functionality. Compared to manual analysis of user feedback, an automated software usage analysis tool works more efficiently for companies with large software development projects. In addition, the traditional feedback recoding mechanism cannot detect and identify all typical usage patterns in the software. Therefore, it is necessary to provide developers with a new tool to analyse software functionality. Furthermore, such a tool should give the companies suggestions for software improvement. With a better understanding of end-user behaviours and software usage, developers can remove low-efficiency or unresponsive features that no one uses or likes. In addition, they can identify areas that are used frequently (and may need a further extension). Finally, they can test whether new features are used for the targeted user segments.

1.3 Discover a New Method of Creating User Profiles

User profiles can reveal the similarities between users and promote products and features users are interested in (Pachidi et al., 2014). User profiling can also illustrate user navigation behaviour patterns and show users' frequently visited features. Several data types can contribute to user profiling in web usage mining, including users' demographic information, web usage data stored in log files (Cooley et al., 1997), and user clickstreams (Wang et al., 2013). Nasraoui et al. (2007) presented a profiling method by summarizing user sessions clustering results in customer relationship management software. It also includes business process information of users to enrich the user profiles. This paper will develop a new method for creating user profiles from software use case models. We will identify differences between user groups in the context of software usage and user interactions and present them using the same models that were used to develop the software in the first place.

1.4 Research Questions

We pose three research questions that summarize the research goals of this paper.

Research Question 1: How can real-time software usage information be collected and represented in a use case based format for analysis and decision making on software evolution?

Research Question 2: How can a software product's use case model be utilised in developing a user profiling strategy?

Research Question 3: Which cluster analysis techniques generate the most effective outcomes and suit the task of use case based user profiling?

In order to address these questions, we propose a novel solution for extracting information from software operational data, using UML models of the software product in user profiling, and obtaining insightful suggestions from cluster analysis.

1.5 Research Overview

This research paper presents a new methodology for studying and specifying unique user types of software products. We follow the design science approach Hevner & Chatterjee (2004) introduced to construct the test object and analysis environment. A software platform is created that consists of user input simulation functions, data collection mechanisms, and data analysis tools. The platform's task is to verify the feasibility of the user profiling method based on use cases. In data analysis, two machine learning algorithms are introduced, k-means clustering and agglomerative hierarchical clustering. In the final chapter, we will compare these two clustering algorithms from different points of view to analyse which algorithm is most suited for finding unique user types in the product management context.

This first chapter introduces this research, stating the research problem and objectives. The second chapter presents relevant literature and knowledge. Several closely related areas will be discussed, including data mining and web usage mining. We will also discuss a wide range of data mining strategies. We also introduce the principle of K-means clustering and the Hierarchical clustering method.

Chapter three presents the methodology we use to apply use case models to software user behaviour analysis. It is able to group software users into unique user types based on the software use case model. The fourth chapter presents an experiment demonstrating the application of use case based user profiling in revealing software usage. This experiment uses

a dataset retrieved from a bank's help desk system. It contains event logs that record interactions between the help desk agent and the company's employees. The outcome is a series of infographic use case diagrams showing how the core feature of the system, logging incidents, deals with various situations.

The fifth chapter presents the use case based user profiling experiment setup. We performed five parallel cluster analyses with the simulated data. They are:

1. K-means clustering with duration domain usage data
2. Agglomerative hierarchical clustering with duration domain usage data
3. K-means clustering with frequency domain usage data
4. Agglomerative hierarchical clustering with frequency domain usage data
5. Agglomerative hierarchical clustering with frequency domain usage data for subtype study

The result should prove the competence of our method in learning user behaviour patterns from the use case aspect. We compare the results in detail afterwards. Five clustering results are compared and discussed. We present the clustering result in tables graphs and use case diagrams. You can use the infographic use case diagram to see the distribution of user types. We also demonstrate a method for discovering more user types using an agglomerative hierarchical clustering algorithm. We compare k-means clustering and hierarchical clustering qualitatively and quantitatively. We can decide which algorithm is best suited for task-based or use case-based clustering of users in product management.

The sixth chapter presents a design for an automated solution to support our user profiling system. We discuss the possibility of achieving a higher level of automated user profiling.

In the last chapter, we draw conclusions from the research project conducted in this thesis. An overview is given of the results and an outlook on the future work of use case-based user profiling.

2 Background and Related Work

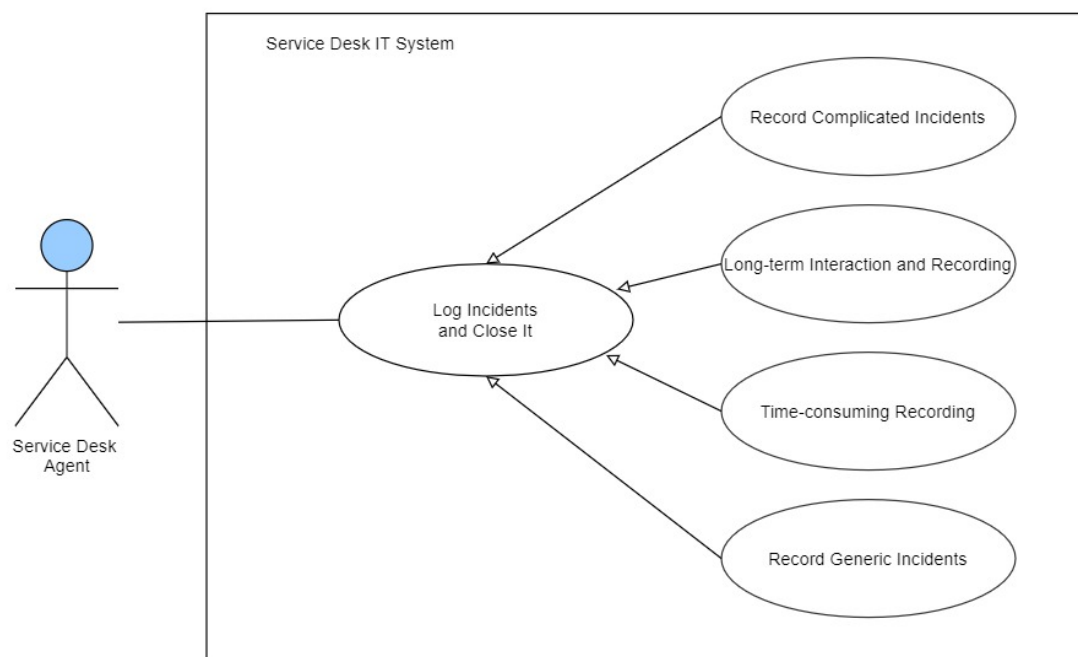
This chapter presents background knowledge relevant to our study. Several areas of information technology are closely related to our research project, including UML diagrams, data mining, especially operation data mining, web usage mining, user profiling, and clustering algorithms. In the following sections, a brief introduction to each knowledge area is provided

2.1 Use Case Models

The Unified Modeling Language was developed between 1994 and 1995 (Booch, 2005). It is a method for system design in object-oriented software development. Use case diagram belongs to the category of behavioural UML diagrams in UML models (OMG, 2007). Use Case Diagrams explain the process of how users interact with the various features of the software. They provide a high-level overview of the structure of the software from the perspective of the user. The use case model can be used to review the specification, architecture, and user interactions (Qazi et al., 2016). Dobing & Parsons (2006) surveyed companies and organizations to understand the situations in which they use UML. They found that use case diagrams are commonly used by organizations. In addition, use case diagrams and activity diagrams are more attractive to enterprise customers.

In a use case diagram, there are components such as actors, use cases, relationships, and system boundaries. Actors are the people (or systems) who interact with the system under development. A use case is a goal that the actors are trying to achieve. Use cases are represented by oval shapes with text annotations. Relationships connect actors and use cases. A simple use case diagram is listed in figure 1. Figure 1 is the use case diagram for the service desk IT system at a bank. There is one actor, 'Service Desk Agent,' and one main use case, 'Log Incidents and Close it.' A subset of four use cases is included in the main use case. Each one represents an incident recording level.

Figure 1 An Example of a Use Case Diagram



There are three types of connections between use cases: include, extend, and abstract. 'Include' is used when primary use cases are constructed. It links primary use cases to their use case subsets. 'Extend' adds optional use cases to primary use cases. 'Subtyping' links use cases to an abstract use case.

In this research, we augment the visualization of use case diagrams to display run-time usage information. Our solution is to add statistical data and colours to the images. By enriching the information in the use cases and relationships, developers and customers should inspect the software system more efficiently.

2.2 Software Usage Data Mining

Usage data refers to the activities that occur when users operate software products. Such a sequence of activities can also be referred to as 'interaction traces' (El-Ramly & Stroulia, E. 2004). Software usage data can be recorded to study the performance of software products. Junco's (2013) research compared two different approaches to measuring Facebook usage by college students. It compared self-reported data with actual usage data collected by software monitoring tools. The experiments showed that by analysing software usage data, we can understand which features are used most often. Usage data also contains information that can give us clues to improve certain features in the software.

Collecting software usage data helps us gain insights into the daily behaviour of end-users and their areas of interest (Germanakos et al., 2008). From a marketing perspective, software usage data is precious because it allows personalized content and promotions to be delivered to end-users. This would increase the likelihood of attracting potential customers. Consequently, the company's overall sales and revenue increase.

Pachidi (2014) introduced three categories of data that need to be collected for operation data mining: Usage Data, User Data, and Enterprise Data. Usage data is the software runtime data that indicates how the system interacts with end-users. Pachidi (2014) listed 18 variables for operational data, including customer ID, user ID, IP address, web server information, database information, application, page, method, function, button triggered, time information, session ID, duration, query duration, error, background tasks, number of records loaded. User data is the demographic data that describes the characteristics of a user. This data can be explicitly recorded in the database by the users themselves. We will also implicitly record user data when we define our own profiling protocol based on the software UML models. Company data is information about the user's organization, including the name and size of the company. What license they own and how many products they have purchased.

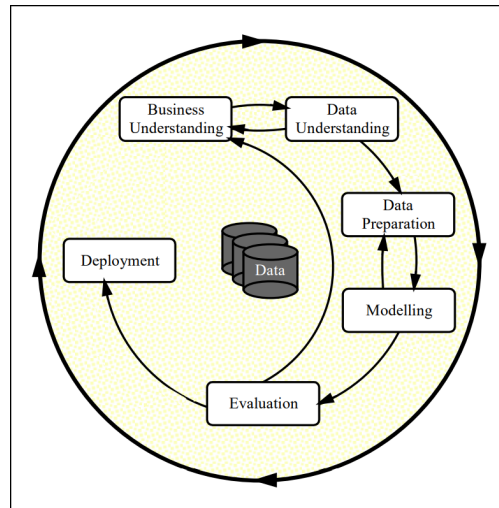
In addition to the low-level software usage data mentioned above, other high-level data should also be considered. High-level data takes different forms for different software systems. In the work of Alexander (2008), menu bar activity in Windows systems is regarded as an important metric for monitoring user behaviour. We use WordPress for our research, such that more contents of the web applications are high-level data to record. For example, navigation bar activity, cookies, etc.

To get useful insights from the raw data, data mining tools provide important mechanisms. Data mining is a technology that can be used for analysing the usage data retrieved from the software system and predicting the patterns contained in the information. We can learn how well the use cases of the system work on the user side by evaluating the interaction data between the user and the system (El-Ramly et al., 2002).

The CRISP-DM model is a widely used classic model for organizing a data mining project. Wirth and Hipp (2000) provided a detailed overview of the CRISP-DM model in their article.

The reference model CRISP-DM presents a six-step life cycle for a data mining project. Figure 2 refers to the CRISP-DM model Wirth and Hipp (2000) used in their article. In the CRISP-DM model, a data mining project includes the following phases: Business understanding, data understanding, data preparation, modelling, evaluation, and deployment. We will refer to the model as an outline for building the data mining prototype in our project.

Figure 2 Phases of the Current CRISP-DM Process Model for Data Mining



Note. From Wirth, R., & Hipp, J. (2000, April). CRISP-DM: Towards a standard process model for data mining. In Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining (Vol. 1).

2.3 User profiling

To more accurately identify user preferences and predict their actions, we need effective user profiling methods. Without proper profiling methods, it is impossible to deliver the right tailored content to end-users as part of marketing. Good profiling methods can help us understand the behavioural patterns and functional requirements of end-users.

Profiling is the process of creating a model that groups users according to their interests and behaviour patterns (Kanoje et al., 2015). There are two primary profiling methods, static profiling and dynamic profiling (Poo et al., 2003). In static profiling, we rely on static attributes and values that users provide themselves to predict their interests. Such data can be provided by end-users through forms and surveys. Moreover, the values are not changed until end users manually enter new data to replace them. Therefore, conclusions drawn from static profiling methods may be inaccurate and outdated.

Another approach to profiling is dynamic profiling. It predicts users' interests based on their actions and digital footprints. We need to capture the user's activities while they are operating the software or website. With the dynamic profiling method, we can quickly find out the current area of interest of the end-user. Such a method is superior to static profiling as it allows more accurate predictions.

User profiling is a handy tool and has a wide range of applications. User profiling and user classification in social media can classify users into different user groups based on their characteristics and activities (Cheng et al., 2010). With dynamic user profiling methods, a social media company can do better advertising and personalization for these users.

In this study, we aim to perform user profiling in regular software products. Our objective is to develop a framework in which the software can recognize the interests of end-users and provide them with personalized features. In an ideal scenario, software users are divided into groups based on their software using behaviours. Users recognized as power users operate the software frequently, and they access most software features. These users require software features that assist them in finishing their professional jobs.

On the contrary, novices that are fresh to the software only use basic features and handle simple tasks. The software should effectively guide them to the basic features they want. People can hide more professional features from novices. The following section presents the cluster analysis data mining techniques that we use in our research project.

2.4 Cluster Analysis Techniques

In cluster analysis, objects are grouped so that they are similar within the same group and different from those in other groups (Kaufman & Rousseeuw, 2009). With cluster analysis, we can complete the task of grouping end users according to their behaviour. This study uses two classical algorithms for cluster analysis: k-means clustering and agglomerative hierarchical clustering. The following section will first introduce these two cluster analysis methods. Then, we will provide information on how to prepare the raw data using normalization and standardization methods.

2.4.1 K-means Clustering Algorithm and Hierarchical Clustering Algorithm Introduction

Unsupervised machine learning is a type of machine learning that works without supervision from users. Unlike supervised machine learning, there is no ideal solution to an unsupervised machine learning problem (Gentleman & Carey, 2008). This research project mainly focuses on two typical cluster analysis algorithms, hierarchical clustering, and k-means clustering.

In hierarchical clustering, there are two different approaches. One is agglomerative hierarchical clustering, and the other is divisive hierarchical clustering. In agglomerative hierarchical clustering, the algorithm considers each data point as a cluster at the beginning. It then combines the clusters with the shortest distance to form large clusters. When only one cluster is left, the algorithm stops. Such a transformation process can be represented with a dendrogram. Divisive hierarchical clustering works in the opposite way. When the algorithm is started, there is one cluster that includes all data points. This cluster is divided into smaller clusters until each data point represents a single cluster.

The K-means clustering algorithm was introduced by MacQueen (1967), and its standard process was defined by Lloyd and published in 1982. It is a partitioning clustering method that divides data points into different groups. In K-means clustering, data points are first assigned to the cluster where the Euclidean distance to the cluster centroid is the smallest. Then the centroid of the cluster is recalculated. This process can be repeated several times. The quality of the clustering result is calculated by the sum of squared errors (abbreviated SSE). If a clustering result has the lowest SSE among all results with the same input and parameters, such a clustering result can be considered as an optimal solution.

During initialization, the algorithm must locate several centroids depending on the number of target clusters. The initial centres can be chosen randomly or from the middle of the dataset. Therefore, K-means algorithms require the user to have a good knowledge of the characteristics of the dataset.

In K-means cluster analysis, the 'Elbow Method' is frequently employed to identify the number of clusters. It operates by locating a graph elbow point where SSE is a function of the number of clusters. As the number of clusters rises, the rate of summation of the distance from each point to the centres begins to drop at the elbow point.

2.4.2 Data Normalization and Standardization

Cluster analysis describes the differences between groups in quantitative approaches. Therefore, data rescaling with normalization and standardization is necessary and recommended (Trebuňa et al., 2014). The process of data normalization and standardization can significantly affect the outcome of cluster analysis. Mohamad and Usman (2013) proved that we could obtain higher quality clusters if we performed data normalization and standardization during data preparation. There are no regulations that force researchers to apply certain normalization or standardization techniques before clustering data. In the following section, we will introduce Z-score standardization and min-max normalization, as these two methods are widely used and representative. In our research project, we chose the min-max normalization method to process the data before cluster analysis.

In a large dataset, standardization can be used to limit the influence of data variability on the cluster analysis results (Tanioka & Yadohisa, 2012). Standardization can be applied when we do not know the maximum value or the minimum value in the data set. It usually leads to better results when the raw dataset follows a Gaussian distribution. Standardization can be applied to both hierarchical cluster analysis and k-means cluster analysis.

The Z-score is a popular standardization method in which attributes are rescaled by the mean and standard deviation. After the standardization process, the mean of the rescaled data is equal to 0, and the standard deviation is equal to 1 (Bhandari, 2021).

For a one-dimensional dataset, the rescaled attribute is calculated by the following formula:

$$X' = \frac{X - \bar{x}}{\sigma}$$

Where \bar{x} is the mean value of the raw dataset, σ is the standard deviation of the raw dataset.

Normalization is a method of rescaling data when the limits of the data set are fixed, and we can apply normalization methods regardless of the distribution type. We can also use normalization in preprocessing data that follows a multimodal distribution. The normalization method we use in this research is min-max normalization. In this procedure, attributes are rescaled to values between 0 and 1. The minimum value of the original data is rescaled to 0, and the maximum value is rescaled to 1.

In Min-max normalization, a one-dimensional dataset will be rescaled according to the following formula:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Where X_{min} is the minimum value in the raw dataset, and X_{max} is the maximum value of the raw dataset.

2.5 Chapter Summary

In this chapter, we reviewed related work on the use of use case models, the organization of data mining projects, and the application of appropriate clustering algorithms. In the next chapter, we review the methodology for building user profiles based on use cases for software products. This use case-based user profiling methodology is based on the application of these techniques. We can successfully make an analysis framework that profiles end-users based on their software system usage activities with the knowledge of software usage data collection, user profiling, and cluster analysis.

3 Use Case Based User Profiling Process

In this chapter, we present the method underlying case-based user profiling. This method aims to improve the quality and efficiency of software product management. It can help group software end-users according to their usage patterns. It also reflects the results of user grouping and software usage conditions in a use case diagram.

The outcome contains infographic use case diagrams that illustrate software usage conditions, as well as suggested improvements to optimize system functionalities. We will examine the result to see if the methodology meets the experiments' requirements. Then we can apply this methodology to an application in a simulated software engineering environment for further investigation.

3.1 Use Case Based User Profiling Process Design

There is not much evidence of using use case models in data mining projects. Mostly use case models are used to illustrate the structure of software. Use case models are comprehensive compared to descriptive sentences. By reading use case models, one can easily understand how actors interact with the system.

The ability of use case models in visualizing software structures at an abstract level is crucial. We want to augment use case models with software usage data through annotation. The annotations will be given various colours reflecting user profiles. In developing the use case based user profiling process, we refer to standard data mining projects. Like the CRISP-DM model (Wirth & Hipp, 2000), our user profiling process consists of four phases. Figure 3 shows the four phases of use case based user profiling process.

The first phase is business understanding. From a software product management point of view, such a process is used to obtain information about software feature usage and end-user preferences in an efficient way. The software usage diagram and its runtime data are the source material in this scenario. In the second phase of data preparation, we need to transfer this data into a dataset of user usage behaviour. The third phase is to analyse this data. We perform a cluster analysis to group the users based on this data set. The last phase is to interpret the result in the form of an infographic use case diagram. We can explain the

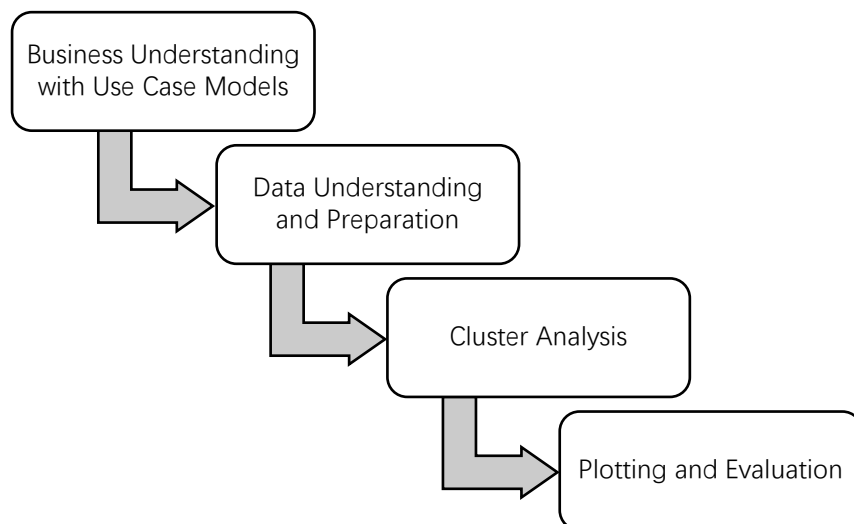
meaning of the diagram and make suggestions to improve the software functionalities.

For applying the use case based user profiling process to analysing the user type partition in a software application, we define a standard clustering analysis process including the following steps:

1. Software use case analysis (identify primary use cases and following use cases)
2. Define user profiling goals (find primary user types or subsequent user types) and select appropriate clustering methods
3. Prepare the usage data dataset
4. Execute the clustering algorithm
5. Calculate statistic data for user types
6. Create infographic use case diagrams
7. Give software improvement suggestions

People may obtain a user type distribution by executing the aforementioned clustering analysis process. Such a user type distribution explains the software usage condition of different user groups from a use case perspective. We will introduce the four phases of use case based user profiling process in detail in the following sections.

Figure 3 Use Case Based User Profiling Process Model



3.1.1 Business Understanding with Use Case Models

In business understanding with use case models, we focus on mapping the critical use cases that reflect end-user software using behaviour. Since the outcome is based on use case diagrams, it is important that a use case diagram represents well the end user's interaction

with the software system. Thus, the most important task in understanding business processes with use case models is to master the structure of the target software and create a suitable use case model. We can also create an appropriate use case diagram for a system if we do not have access to its use case diagram. This may take more time but provides more flexibility in customizing the use case structure.

3.1.2 Data Understanding and Preparation

Data understanding and preparation is the process of collecting appropriate data and transforming that data into a manageable data set for cluster analysis by the user. The data must be relevant to specific use cases in the use case model, i.e., this data is generated by the features included in a use case. We focus on capturing and analysing software usage data in the use case based user profiling process. Usage data is low-level data that contains information about the operation of the software. We consider the following data to be the most important in user profiling: user identity information, activity information, activity duration, and the number of records loaded. These variables indicate not only which functionalities are enabled but also how long and how often they are used. Thus, we can easily assign these variables to specific use cases during cluster analysis.

In an ideal scenario where people are applying the use case based user profiling process, they should first select a range of software usage data in the system that can be mapped to use cases in the use case model. These usage data may represent how long and how frequent end users are interacting with specific use cases. Then, people may define a period in which to record these usage data. After this period, they can collect the data and export a manageable data set in CSV format or others.

3.1.3 Cluster Analysis

Cluster analysis divides users into heterogeneous groups. Within each cluster, all users are similar in their software usage behaviour. In other words, the differences between usage data values within one user group are smaller than those among groups. The clustering algorithm we choose to accomplish this task is k-means clustering and agglomerative hierarchical clustering. Users' software usage data are the input values in the clustering process.

3.1.4 *Plotting and Evaluation*

The result of the cluster analysis yields a distribution of user types. The next task is to map this distribution into a use case diagram and interpret the result in terms of recommendations for improving the software. An infographic use case diagram is built for each user type to map the distribution in a use case diagram. Infographic use case diagrams are annotated use case diagrams, including descriptive text and coloured shapes. The descriptive text infers the statistic data of the clustering results. The use cases in the infographic use case diagrams have different colours representing how users of a specific type interact with use cases on average. People can define a colour scheme to present user profiles. For example, they can define dark colours in a use case eclipse representing an intensive usage. The darker the colour of the use case is, the more intensive they are invoked by end-users.

We may investigate the differences between infographic use case diagrams of various user types to make decent software improvement suggestions. For example, there is a 'power users' type whose infographic use case diagram has use cases of dark colours. The dark colours indicate that 'power users' often access these use cases. We suggest that the software optimize the durability of the features in these use cases for the 'power users' since they never expect software failures during hard work. Of course, this evaluation process has to be performed by a human.

3.2 Chapter Summary

In this chapter, we illustrate the use case based user profiling process in detail. We can discover different user types from cluster analyses using the use case model as a high-level design. The software improvement suggestions may be drawn according to the infographic use case diagrams, hindering and revealing. Software developers and product managers can apply our profiling process to improve the software in the next iteration of the software functions.

4 Use Case Based User Profiling in a Help Desk Interaction System

The Help Desk Interaction System cluster analysis experiment is designed to showcase the use case based user profiling in clustering user behaviours in a software product. This experiment presents the method of creating an infographic use case diagram reflecting specific features usage. The experiment procedure follows the standard use case based user profiling process. In the beginning, we will study the software system's structure to create a suitable use case diagram. Then, we will select several categories of variables from the dataset that are relevant to system core features for cluster analysis. The experiment applies K-means clustering to group the incident logs into unique types. Infographic use case diagrams will demonstrate which types of incident recording exist and what use cases can be mapped to them. This experiment uses an open-source dataset of event logs from a bank's service desk for cluster analysis. At the end of the experiment, we give suggestions on improving the service desk software system. We use a dataset of event logs that revolve around the service desk processes in a bank. The dataset is open and uploaded to the website IEEE Task Force On Process Mining, and the URL is

https://www.win.tue.nl/ieeetfpm/doku.php?id=shared:process_mining_logs

The cluster analysis process has seven steps, which are

1. *Prepare the dataset (with or without data normalization).*
2. *Define the cluster number.*
3. *Apply the K-means clustering algorithm.*
4. *Calculate the mean value of attributes of each cluster.*
5. *Map clustering results to use cases.*
6. *Create infographic use case diagrams with clustering results*
7. *Make software improvement suggestions.*

The details of the cluster analysis will be introduced later in this chapter. The following section shows the process of learning the software's system design and use case model.

4.1.1 Help Desk Interaction System Business Understanding

The dataset we use contains information from a help desk interaction system. We do not have access to any materials revealing the system structure. Thus, we have chosen an inductive

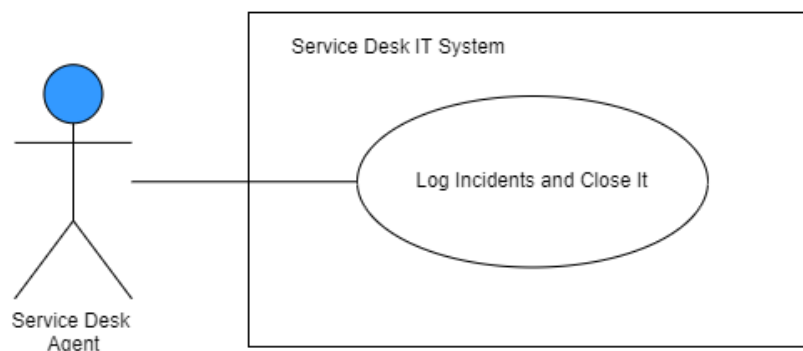
reasoning approach to restore the system use case diagram design. The dataset has 17 types of variables. We present the title of these variables in table 1.

Table 1 Service Desk Dataset Variables Overview

Variable Title:					
CI Name (aff)	CI Type (aff)	CI Subtype (aff)	Service Comp WBS (aff)	Interaction ID	Status
Impact	Urgency	Priority	Category	KM number	Open Time (First Touch)
Close Time	Closure Code	First Call Resolution	Handle Time (secs)	Related Incident	

We assume that all system usage data and user data are included in the dataset. From the contents of the variable categories, we conclude that the only activity that occurs between the end-users and the system is 'log Incidents and Record It.' So there is only one primary use case for this software system, and the actor is the service desk agent. The service desk interaction system records incidents' characteristics with different variables. The 'handling time' category captures the duration of the resolution of the incident. 'Impact', 'Urgency' and 'Priority' categories indicate the importance of handling specific incidents. So there should be more use cases that indicate the level of handling incidents. We will first apply the clustering algorithm to group incidents into heterogeneous groups. The cluster analyses result should tell how many unique incident handling types exist. Figure 4 is the primary use case diagram of the service desk interaction system. One actor, the service desk agent, interacts with the primary use case, 'log incidents and close it.'

Figure 4 Service Desk IT System Use Case Diagram



Since the current use case diagram is too simple to reveal the software usage situation, the goal of the clustering process in this scenario is to enrich the use case diagram of the service desk system. First, we will cluster these incident logs into unique groups reflecting how the agent copes with different task loads. Each group is mapped to a use case representing how the agent uses the system to handle a specific type of incident. Then we add these use cases to the use case diagram. Finally, we will annotate the cluster analysis results in the infographic use case diagrams, and make software improvement suggestions.

4.1.2 Data Preparation

The original dataset has 17 variable categories. Most of the variables are irrelevant to the cluster analysis task. The condition filtering the variables is that these variables should present information on event types, incident urgency, and interaction time cost. The variables should also reflect the character of each incident. We limit variable categories to seven. The variables we selected belong to the following categories, 'CI Type (aff)', 'CI Subtype (aff)', 'Closure Code', 'Impact', 'Urgency', 'Priority', 'Handle Time (secs)'.

We select 'CI Type (aff)' 'CI Subtype (aff)' values since these values may indicate the type of hardware devices affected by the incidents, which should help differentiate incidents. The 'Impact,' 'Urgency,' 'Priority,' and 'Handle Time (secs)' may infer the importance and complexity of the incidents handling processes. We can use these variables as input in the cluster analyses process. A new data set is made with these seven variables. Table 2 shows the first ten rows of the data as a preview. The data set contains 147004 rows of values. Each row represents an event recorded by the helpdesk agent.

Table 2 Service Desk IT System Dataset Overview

	CI Type (aff)	CI Subtype (aff)	Impact	Urgency	Priority	Closure Code	Handle Time (secs)
0	application	Server Based Application	5	5	5	No error - works as designed	98
1	#N/B	#N/B	5	5	5	Other	235
2	software	Automation Software	5	5	5	Referred	239
3	software	Automation Software	5	5	5	Software	498
4	computer	Banking Device	5	5	5	No error - works as designed	597
5	computer	Banking Device	5	5	5	Other	317
6	computer	Banking Device	5	5	5	Software	430
7	computer	Banking Device	5	5	5	Operator error	884
8	computer	Banking Device	5	5	5	Other	529
9	computer	Banking Device	4	5	4	Data	670

4.1.3 K-means Cluster Analysis without Data Normalization

The first clustering process doesn't contain a data normalization. We remain this clustering analysis and its results because the differentiates logs by how long they last until closed

When performing the K-means clustering algorithm without data normalization or standardization, the clustering results may be affected by the large gap in values between 'Handle Time' and other variables. This is because the standard deviation of the variable 'Handle Time' is much larger than that of the other variables. Therefore, we anticipate that the clusters in the result are differentiated primarily in their 'Handle Time' values.

We select values from the 'Impact,' 'Urgency,' 'Priority,' and 'Handle Time' columns as input values since they are numeric and may reflect incident characteristics. These numeric fields can be easily used as input values in K-means clustering.

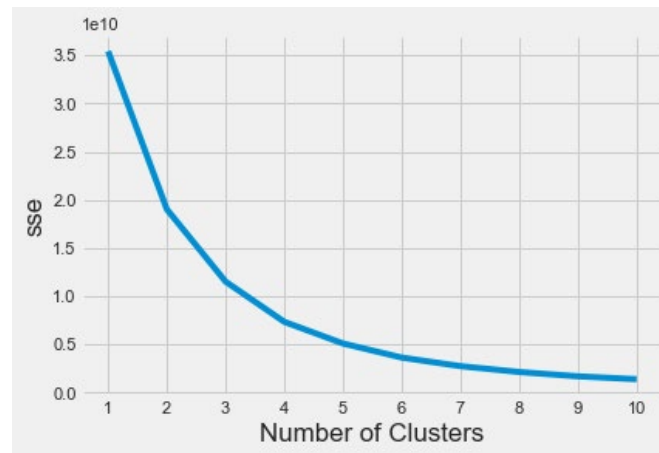
The first step in K-means clustering is to specify the number of clusters we want to generate in the K-means cluster analysis. Then, we apply the 'Elbow Method' to check which number of clusters is an appropriate arrangement.

4.1.3.1 Determine Cluster Numbers by Elbow Method

The 'Elbow Method' is often used to determine the number of clusters in K-means cluster analysis. It works by finding an elbow point in the graph where SSE is the function of the number of clusters. At the elbow point, the rate of summation of the distance from each point to the centres begins to decrease as the number of clusters increases.

The picture of the setting of SSE as a function of the number of clusters is shown in Figure 5. Figure 5 shows the function of SSE to the number of clusters. In this case, the elbow point is at a number of 4 clusters. Therefore we decide to set the number of clusters to four in the K-means cluster analysis.

Figure 5 SSE as a Function of the Number of Clusters



4.1.3.2 K-means Clustering Result

The centroid of each cluster is shown in Table 3. Since we have four input values, the centroid of each cluster also has four attributes. As we can see, these centroids differ mainly in the time field, which represents the total cost of resolving an incident in the helpdesk.

Table 3 Service Desk K-means Clustering Centroids Details

	<i>Centroid 1</i>	<i>Centroid 2</i>	<i>Centroid 3</i>	<i>Centroid 4</i>
<i>Impact</i>	4.25	4.24	4.29	4.25
<i>Urgency</i>	4.25	4.24	4.29	4.24
<i>Priority</i>	4.23	4.23	4.28	4.22
<i>Time</i>	733.79	236.05	7310.56	1775.9

Table 4 demonstrates the key attributes of the clusters. The first row is the number of elements (event logs) included in the cluster. The second row shows the partition each cluster holds. The third row is the average value of the variable 'handle time' in each cluster. We also put the minimum and maximum value of the 'handle time' in each cluster in the table.

Table 4 Cluster Details of the Service Desk Cluster Analysis

	<i>Cluster 1</i>	<i>Cluster 2</i>	<i>Cluster 3</i>	<i>Cluster 4</i>
<i>Counts</i>	39312	101192	195	6305
<i>Proportion</i>	26.74%	68.84%	0.13%	4.29%
<i>Avg Time(s)</i>	733.95	236.05	7310.56	1776.89
<i>Max Time(s)</i>	1254	484	22530	4491
<i>Min Time(s)</i>	485	0	4546	1255

We can come to the conclusion that cluster 2 is the largest cluster, containing 68.84% event logs, and cluster 3 is the smallest, containing only 0.13% event logs. Cluster 3 and cluster 4 are in the middle in terms of volume. The average time cost in cluster 3 is the highest, while the average time cost in cluster 2 is the lowest. The results will be discussed in detail in the following section.

4.1.3.3 Plotting and Evaluation

The plotting and evaluation process is used to interpret the results of the cluster analysis and shows that use case diagrams can be expanded into different forms. Several designs have been tested to represent cluster information in the use case diagram. The selected version is a use case diagram with cluster distribution information within use case eclipses. In addition, a colour-based infographic use case diagram design is introduced in the supplement section at the end of this chapter. Figure 6 shows the infographic use case diagram we made for the service desk interaction system with clustering results.

The primary use case is 'Log Incidents and Close It.' This is the process of using the Service Desk IT system to communicate with company staff and log incidents. Four other use cases are associated with the primary use cases.

Four clusters from the K-means clustering are mapped to four use cases. The first use case, 'Record Complicated Incidents', represents Cluster 1. Cluster 1 has the second largest volume and the second shortest average processing time of all.

The second use case represents Cluster 3, which is the smallest, and its logs have a very long processing time. We assume that these individuals actively communicate with the agent until the incidents are closed. Such an activity type fits well with the means of the title 'Long-term interaction and recording'.

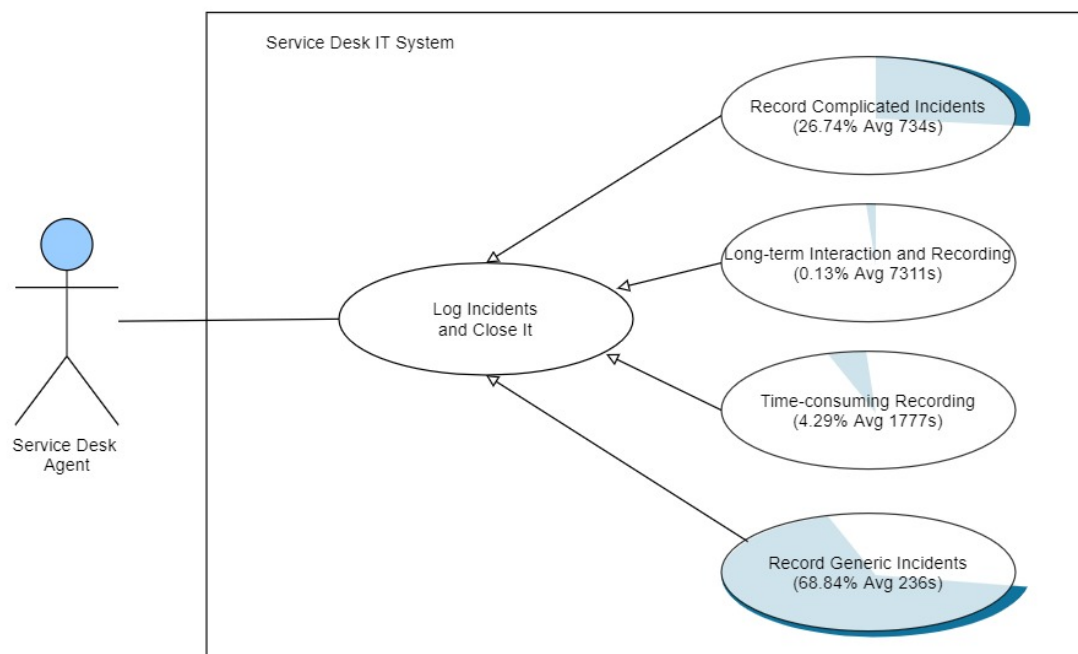
The third use case is 'Time-consuming Recording,' which represents cluster 4. Logs in cluster

4 take more time than in cluster 1 and cluster 2.

The last use case is 'Record Generic Incidents', which represents Cluster 2. Cluster 2 has the highest volume and the shortest average processing time.

The infographic Use Case Diagram demonstrates the percentage each following use case holds in the prime use case. The average processing time for each use case is added to the eclipse. The blue area within each eclipse represents the exact percentage value. The numbers are included in the eclipses so that the viewer can read the statistical information without having to look back at the tables.

Figure 6 Service Desk Infographic Use Case Diagram



From the clustering result, we conclude that service desk agents face four different degrees of workload. The software should provide agents with features that allow them to handle generic incidents well.

More than 90% of interactions last less than 20 minutes. Handling such interactions is the main task for the help desk system IT. The system should ensure sufficient computing power for these communication, recording, and logging activities.

In addition, the system must provide sufficient computing power for long-term interaction activities. Under extreme conditions, an interaction may last several hours. The system should

have memory or buffers to prevent memory overflow. In the future, the improved system can help agents easily track time-consuming calls. The system should accurately record problem-solving processes and keep them in good working order.

4.1.4 K-means Cluster Analysis with Data Normalization

In this cluster analysis, we added data normalization. With the data normalization process, we anticipate clusters would be formed differently. In this second K-means cluster analysis, we apply the 'min-max' approach to normalize the 'Impact,' 'Urgency,' 'Priority,' and 'Handle Time' variables. We set the target cluster number to four to compare the result with K-means clustering without normalization. The result is shown in table 5.

4.1.4.1 K-means Clustering with Data Normalization Result

Table 5 K-means Clustering Result Overview with Normalized Data

	<i>Cluster 1</i>	<i>Cluster 2</i>	<i>Cluster 3</i>	<i>Cluster 4</i>
<i>Counts</i>	52182	77533	949	16340
<i>Proportion</i>	35.5%	52.7%	0.6%	11.1%
<i>Avg Time(s)</i>	436.63	450.16	379.53	448.04
<i>Max Time(s)</i>	22530	18227	2630	13234
<i>Min Time(s)</i>	0	0	0	0

From this result, we notice that four clusters have very close average handle time values with the data normalization process. Cluster 2 is the largest, holding 77533 instances. Cluster 1 is the second largest, holding 52182 instances. Cluster 3 and cluster 4 are relatively small. Cluster 3 is the smallest, holding only 949 instances. Table 6 presents the average value of these clusters 'Impact', 'Urgent,' 'Priority' variables, and the minimum and maximum values in the clusters. We notice that instances in cluster 1 have the highest 'Impact', 'Urgent,' and 'Priority' values with an average of five. Instances in cluster 3 have the lowest 'Impact', 'Urgent,' and 'Priority' values with an average of two.

In summary, these clusters are differentiated in the interaction characteristics rather than handle time. Furthermore, it shows that various inputs in the K-means clustering process generate multiple conclusions. Therefore, we need to understand the dataset before starting the clustering process.

Table 6 Incidents Characteristics in the Clusters

	<i>Cluster 1</i>	<i>Cluster 2</i>	<i>Cluster 3</i>	<i>Cluster 4</i>
<i>Avg Impact</i>	5	4	2	3
<i>Min Impact</i>	5	3	1	3
<i>Max Impact</i>	5	5	4	5
<i>Avg Urgent</i>	5	4	2	3
<i>Min Urgent</i>	5	3	1	1
<i>Max Urgent</i>	5	5	3	4
<i>Avg Priority</i>	5	4	2	3
<i>Min Priority</i>	4	4	1	3
<i>Max Priority</i>	5	4	2	3

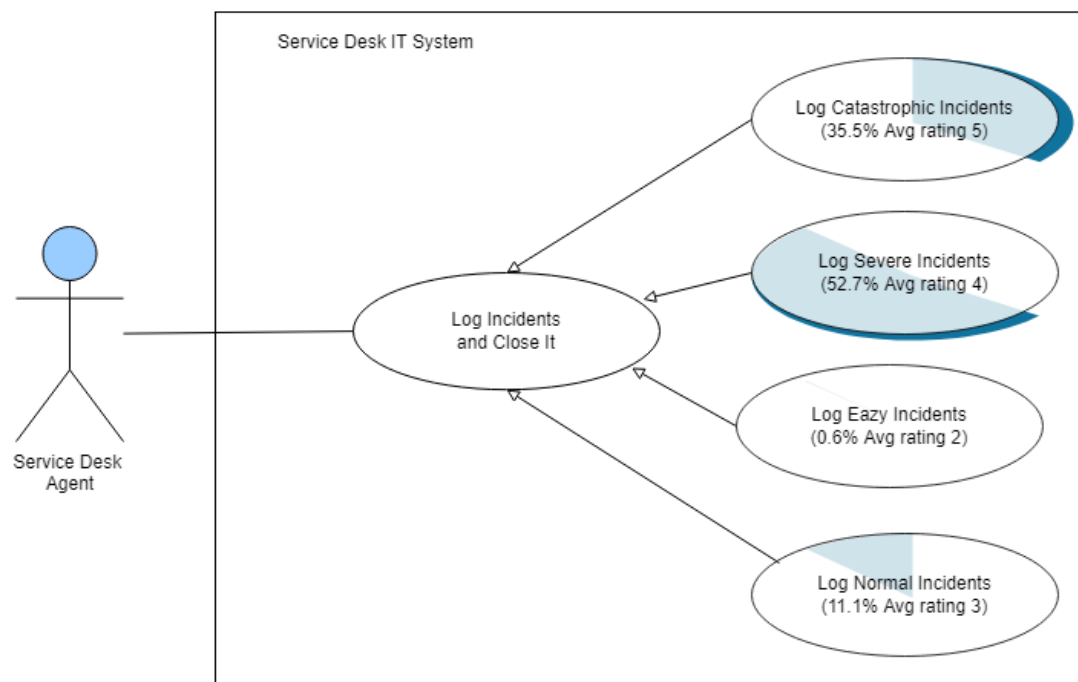
4.1.4.2 Plotting and Evaluation

We make a new infographic use case diagram based on the result from the second K-means clustering analysis. Figure 7 represents the infographic use case diagram with a use case subset including 'Log Catastrophic Incidents,' 'Log Severe Incidents,' 'Log Easy Incidents,' and 'Log Normal Incidents.' We name these four new use cases based on their average 'Impact,' 'Urgent,' 'Priority' ratings. The higher the value, the more severe these incidents are.

We call the first use case 'Log Catastrophic Incidents' since the average value of 'Impact,' 'Urgent,' 'Priority' in its corresponding cluster is the highest. We input its average rating value of five in the use case eclipse. The second use case, 'Log Severe Incidents,' has an average incident influence rating of four. The third one, 'Log Easy Incidents,' has an average rating of two, which is the lowest among all. The fourth one, 'Log Normal Incidents.' has an average rating of three.

Cluster 1 and cluster 2 hold up 88% of the service desk interactions. The infographic use case diagram reveals that these interactions have high 'Impact,' 'Urgent,' 'Priority' values. The service desk agent rated most incidents quite severe. The suggestions are leaving sufficient storage space for recording incidents of powerful influences. The system must ensure that all severe incidents are tracked and solved accordingly.

Figure 7 Service Desk Infographic Use Case Diagram Part 2



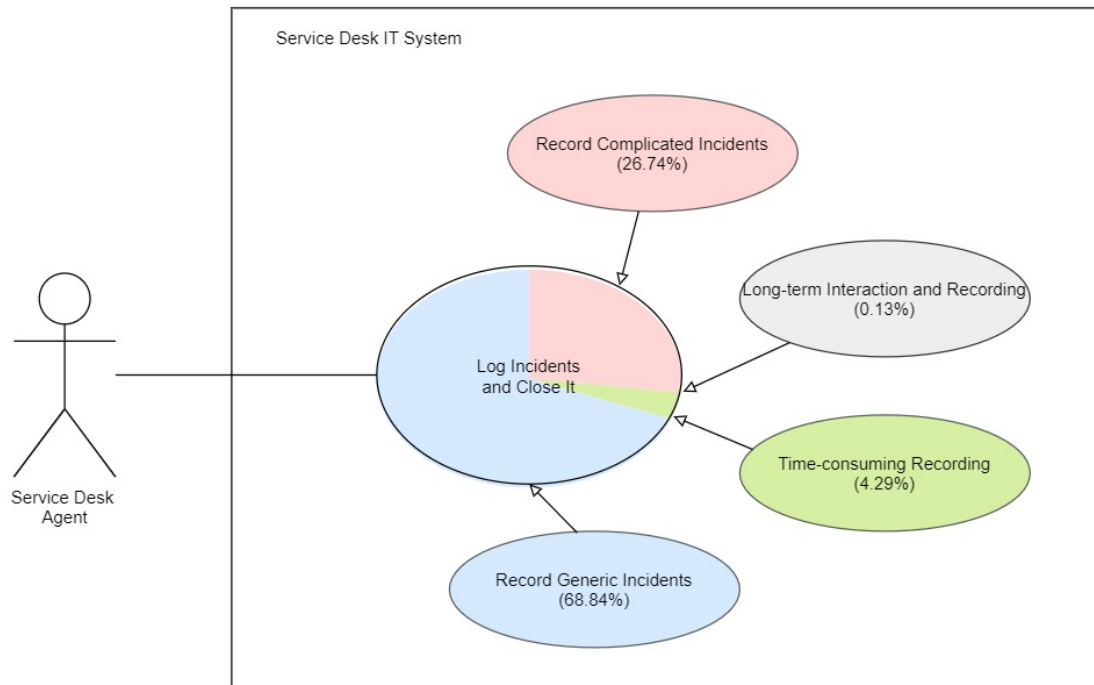
4.2 Chapter Summary

In this chapter, we perform two cluster analyses demonstrating the process of applying use case based user profiling in evaluating software usage and user behaviours. The outcome proves our use case based user profiling can help people replenish the unnoticed following use cases in the software use case model.

4.3 Supplement: A Coloured Infographic Use Case Diagram Design

Use Case Diagram (Ratio Based Infographic Design)

Figure 8 Ratio Based Infographic Use Case Diagram Design



This is another design of the infographic use case diagram. The primary use case is 'Log Incidents and Close It.' The use case field is in the form of a pie chart and coloured in 4 different colours, each representing a specific subtype. The size of the coloured area indicates the importance of each subsequent user type.

Compared to the infographic use case diagram with shapes representing cluster distribution, the colours in the eclipse cannot clearly represent the proportion information. However, it has the potential to represent intensity or other quantitative attributes. Hence, we can use colours in use case diagrams in the prototype for software user profiling.

5 Use Case Based User Profiling in a Web Blog Application

This chapter introduces an experiment of applying the use case based user profiling process to analysing user behaviours in a web application. The experiment aims at evaluating the performance of the use case based user profiling in discovering user types in software applications. We may compare the K-means clustering and hierarchical clustering in terms of profiling users by use cases.

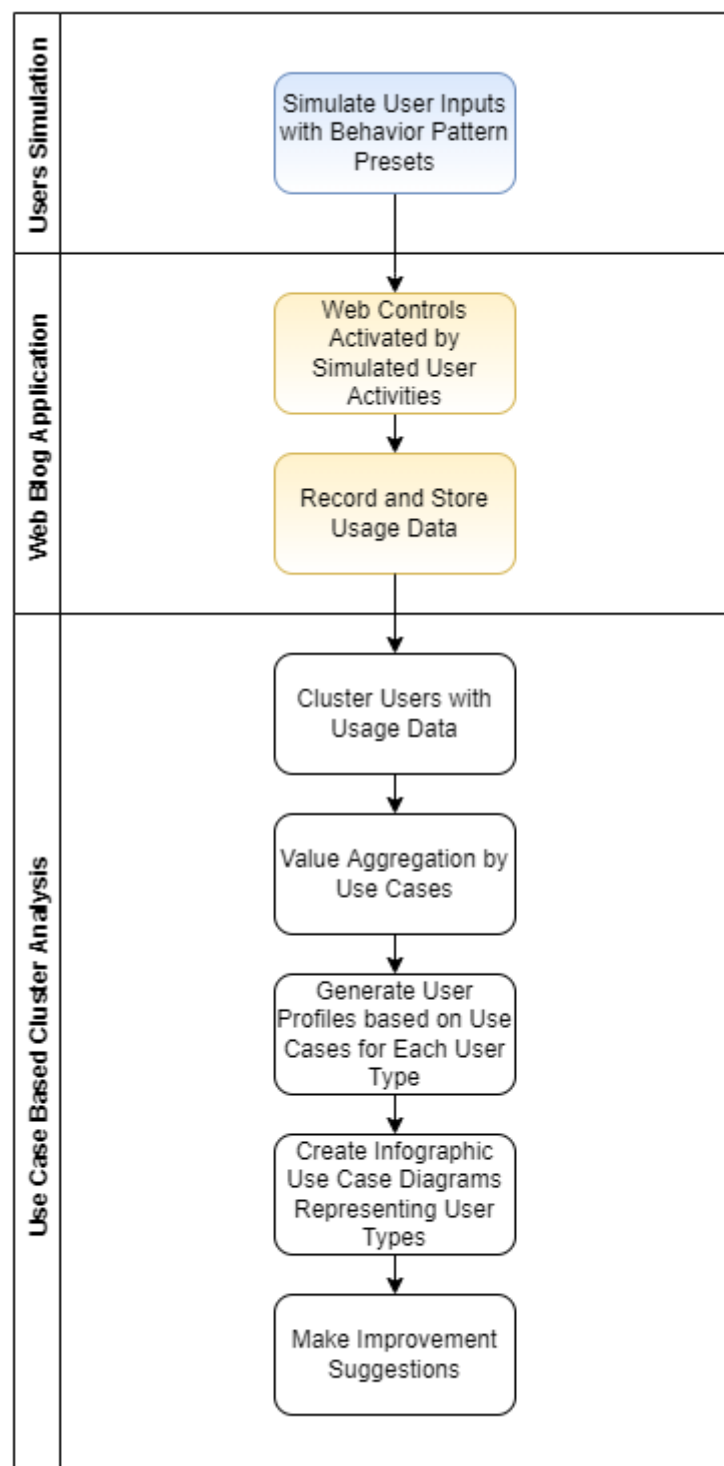
5.1 Software Components in the Experiment

Use case based user profiling process can be applied to the software which documents user usage data. For experiment purposes, we build an application as the test object. Use case based user profiling process is applied to analyse the users of this application. Its usage data are generated by simulated user inputs. Figure 9 shows the experiment workflow structure in the experiment. Three pools represent three primary components that form the entire experiment environment, the web blog application, the user simulation program, and the use case based cluster analysis program. Each software component has its unique job.

The web blog application is a program we developed especially for the experiment. It is a web blog system with features people can commonly see on personal blogs. The web blog application contains features like posting blogs, leaving comments, viewing contact information, etc. The blog system also can record visitors' usage data. The web blog application includes a database that stores the software usage data and visitors' credentials.

The user simulation program is created for generating simulated user inputs. This program activates web controls on the web blog application following predefined scripts. When the web blog application is operated by simulated users and is running, it stores usage data in its database. This way, we can get a set of software usage data ready for cluster analysis.

Figure 9 Experiment Software Components Structure



The use case bases cluster analysis program is written in Python 3, mainly with 'Pandas,' 'Numpy,' and 'Sklearn' libraries. These libraries offer functions organizing datasets and performing cluster analysis. We use both the K-means algorithm and the agglomerative hierarchical algorithm for cluster analysis of the individual datasets. In addition, to

demonstrate the 'subtype discovery' feature of the agglomerative hierarchical algorithm, we apply the hierarchical algorithm twice with different parameters to the cluster analysis of the frequency domain. We follow the steps below to perform the cluster analysis.

1. *Normalize the dataset with min-max normalization.*
2. *Choose the appropriate cluster number.*
3. *Apply the clustering algorithm.*
4. *Calculate the mean value of key attributes of each cluster.*

These steps generate clusters that represent unique user types. The following job is to aggregate values for profiling users by use cases involved. The method we applied in the profiling process to organize usage data is to aggregate numeric values of the variables of the same category. For example, we defined eight variable categories in retrieving usage data from the blog system. The values represent a summation of time or frequencies. Also, in creating user profiles based on use cases, we aggregate duration and frequency values for each use case. We can analyse how much time users interact with each use case or how frequently they interact with these use cases. We follow the steps below to aggregate data.

1. *Map the variables to the blog system use cases.*
2. *Accumulate the variables which map to the same use case for each cluster.*

After data aggregation, we have obtained use case based user profiles and distribution of user types. Then we need to process these data further to visualize the user profiling result. There are five steps in creating the infographic use case diagrams and making improvement suggestions.

1. *Create a new dataset of the duration/frequency domain user profiles*
2. *Normalize the dataset with min-max normalization.*
3. *Calculate the variable mean value.*
4. *Create a heatmap with the mean values produced from the last step.*
5. *Create infographic use case diagrams representing user types*
6. *Making software improvement suggestions.*

These form the procedure of visualizing the critical attributes of each user type. The outcomes include a series of infographic use case diagrams showing users' software using habits and a

series of heatmaps indicating usage intensity and frequency of use cases.

In the execution of the experiment, we test the K-means clustering and hierarchical clustering with the task of analysing software usage data. The user simulations program generates two datasets. One contains software usage data in the duration domain, and the other contains usage data in the frequency. In total, we performed five instinct cluster analyses. They are

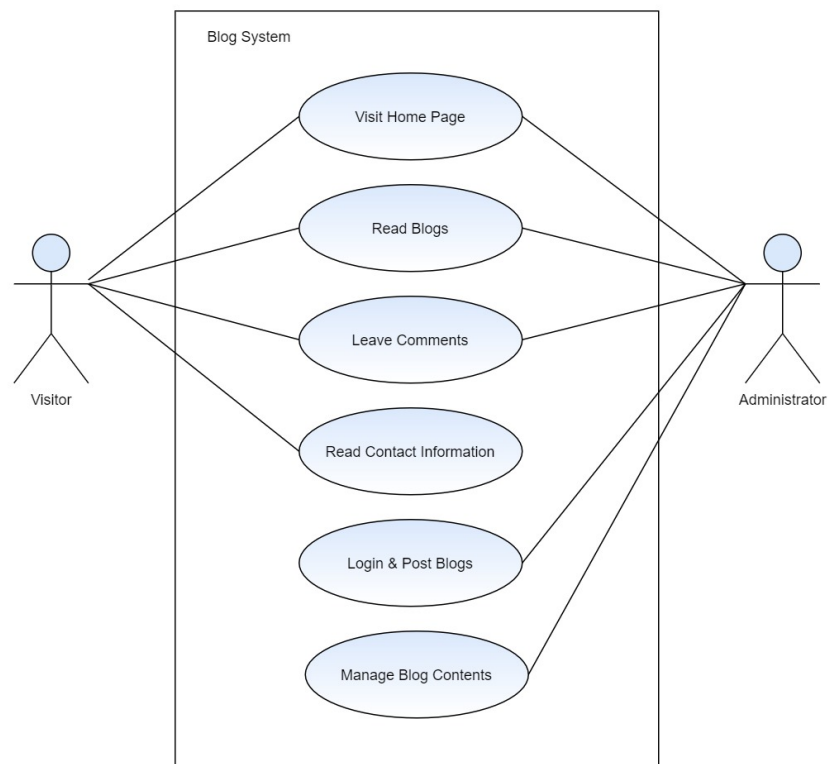
1. K-means clustering with duration domain usage data
2. Agglomerative hierarchical clustering with duration domain usage data
3. K-means clustering with frequency domain usage data
4. Agglomerative hierarchical clustering with frequency domain usage data
5. Agglomerative hierarchical clustering with frequency domain usage data for subtype study

We will introduce the web blog application we build in detail in the following sections. Then we will illustrate how do we design the user simulation strategy. We will also explain how we develop the data aggregation method to map software usage data to use cases. At last, we showcase the use case based cluster analyses processes and corresponding results.

5.2 Web Blog Application – A Blog Management System

The web application is built with WordPress. WordPress is a popular content management system worldwide. The advantage of creating an application with WordPress is that a large number of web usage tracking and data monitoring tools directly support the WordPress platform. Moreover, the templates in WordPress can reduce the overall cost of our project development. Figure 10 shows the draft use case diagram for the blog system. The blog system has two actors, the visitor, and the blog administrators. The visitor is linked with four use cases, visit home page, read blogs, leave comments and read contact information. The administrator is also linked with use cases of visit home page, read blogs and leave comments. However, the blog administrator doesn't need to read contact information to contact himself. The administrator can post blogs on the web blog application and manage blog contents, including removing blogs he doesn't like.

Figure 10 Blog System Use Case Diagram



The web application we developed is a blog. Users can visit the pages on the blog websites. There is the main page, including the welcome information and navigation bars. Figure 11 shows the design of the home page of the website. Visitors can also see a list of existing blogs. By clicking the plus buttons, visitors can easily visit the blog that interests them. Figure 12 shows the text box that navigates to the blog content.

Figure 11 Blog System Homepage

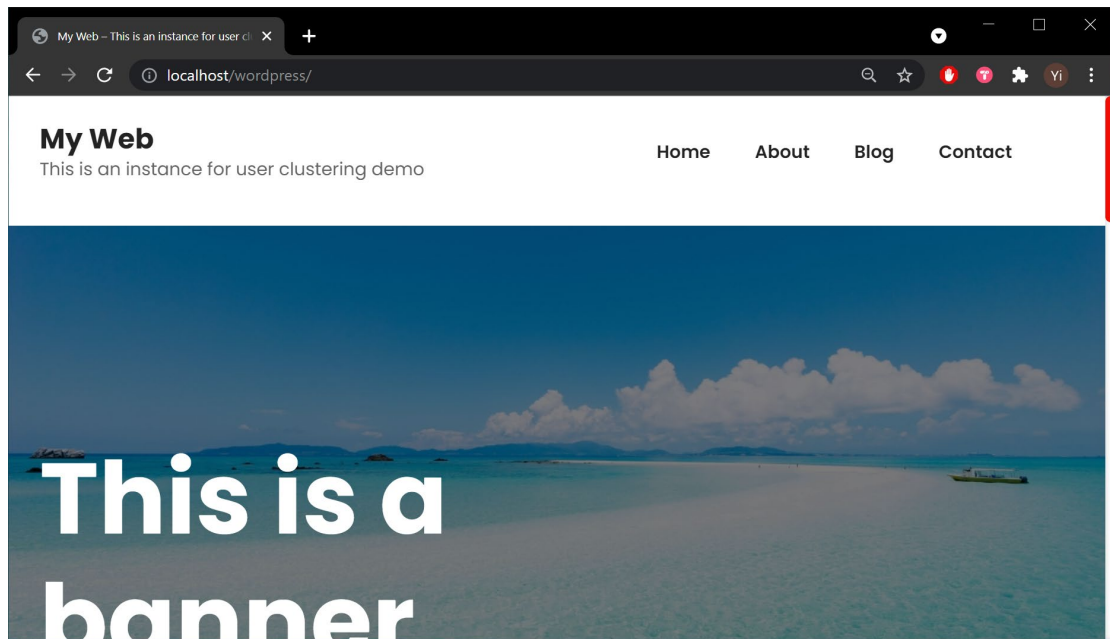
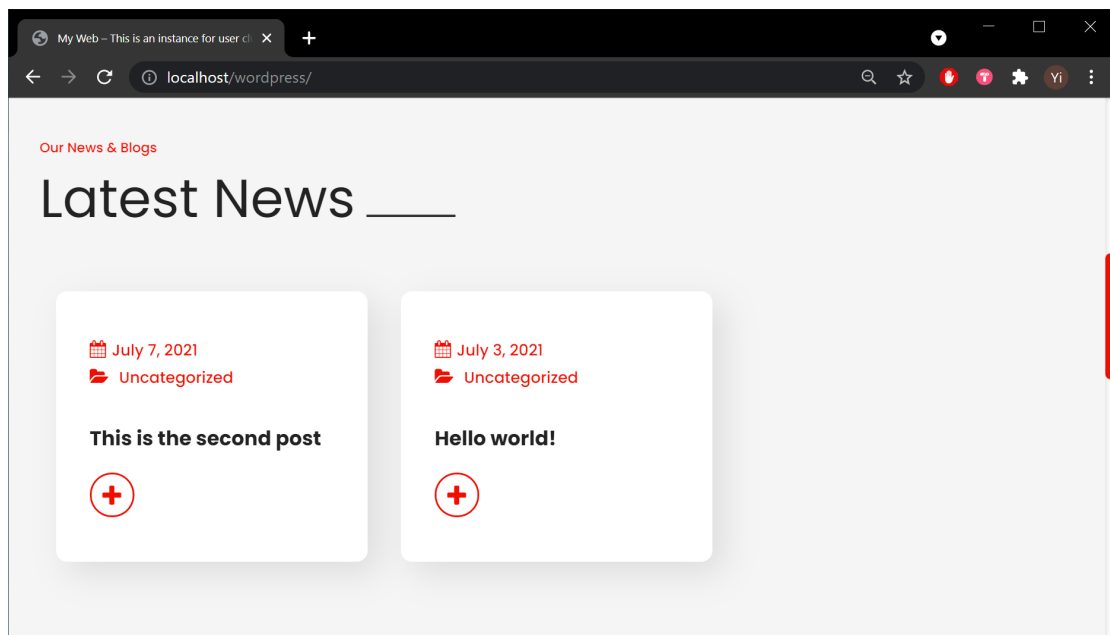


Figure 12 Blog System Blog Content Entrances



Visitors can click on the text box in the navigation bar to access individual pages. The About page provides information about the authors. On the contact page, visitors can read the contact information of the site's authors. When visitors click on a blog entry, they are taken to a page where all blog content is listed. Figure 13 shows the main design of the blog page. First, they can select the blog they want to read. Then they can leave comments on the blogs. Figure 14 shows the design of the blog content page. On the blog page, visitors can see others' comments and leave their comments.

Figure 13 Blog System Blog Content Main Page

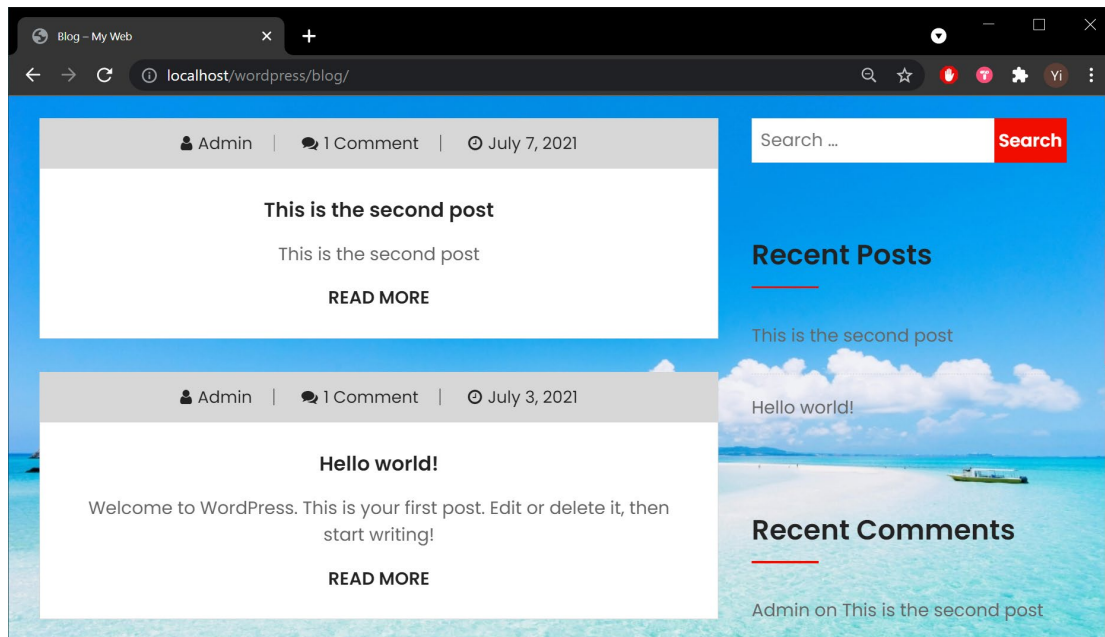
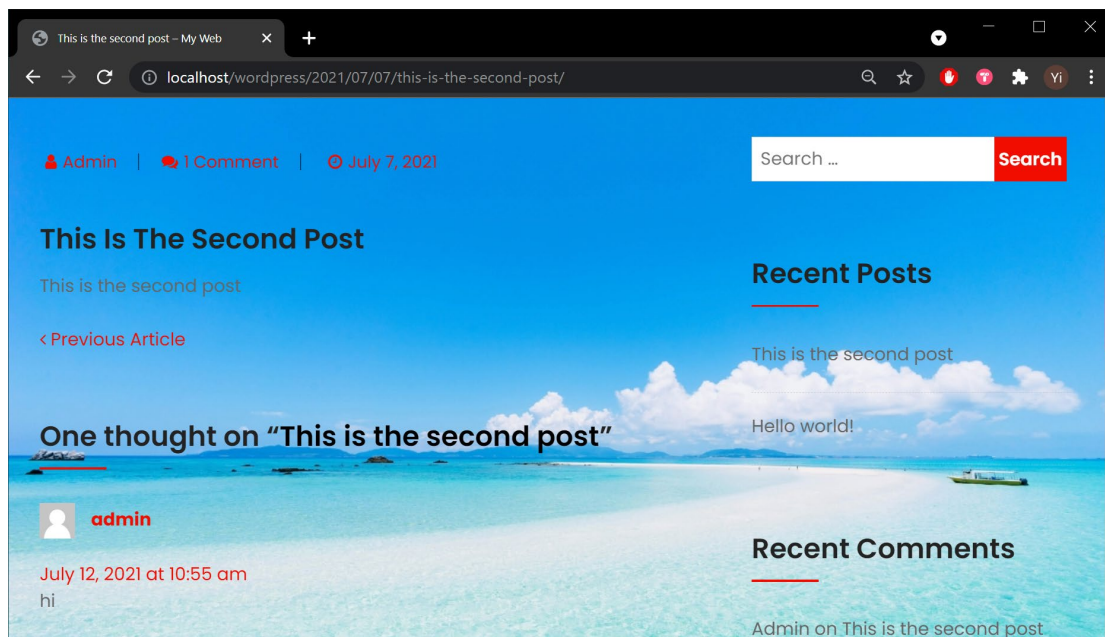


Figure 14 Blog System Blog Page



The high-level design of this blog software is shown in the following use case diagram. Visitors can go to the home page, where they can find general information about the blog. They can view individual blogs and associated comments. They can also leave a comment on the blogs they like or dislike. In addition to reading blogs and comments, visitors can also access the contact information and about page to learn more information about the blog owner. The blog owner can register and publish blogs as an administrator and delete blogs.

5.3 User Input Simulation Method

The use case based user profiling process runs on software usage data. Therefore, we need sufficient software usage data to verify the performance of implementing user profiling in a software application. In our experiment, the web blog application requires many visitors to visit the website to proceed with the user profiling process.

One way to generate usage data is by recruiting test candidates as web blog application users. They may visit the website, read a few blogs, and leave comments. Human users provide more natural input data than test scripts. However, humans can repeat their input in our test to complete their assigned tasks as quickly as possible. Other factors such as Internet connection quality and cultural differences may prevent our candidates from providing reliable input, leading to unreliable results when clustering users.

Another option is creating a test script or software that simulates user input. These 'users' can be either real people or computer programs. Actual users would reflect more closely the application of our methodology. However, within this research's time scale and context, it was not feasible to obtain a statistically significant user base. Therefore, we opted for an approach of user input simulation

The script plays the role of a random user interacting with the system. We specify in advance in the script some typical behaviours of the user to control the output. This method is suitable for checking the model's design and justifying whether the model meets our research requirements. Furthermore, it is elastic; we can change the script if the demand has changed.

5.3.1 User Input Simulation Objective

Creating a test script that simulates user input is more practical and effective in our experiment scenario. The disadvantage of using scripts is that the clustering result will reflect the behaviour patterns we defined in the scripts. We add randomness in the user simulation scripts to reflect the random behaviour of users. We define various classes of users in the user simulation process on purpose. Each class has a unique behaviour pattern depending on its role. For example, we will set the role of 'new users', and very few software features will be activated in this behaviour pattern. The simulation program will generate user input data

across multiple behaviour patterns. Therefore, we can compare the user behaviour clustering result to the predefined patterns, and prove the availability of the user profiling system prototype.

To make the test cover as many user behaviour patterns as possible, we design the user simulation in two different ways. The first user simulation was created for testing user profiling process based on software usage data in the duration domain. The second one is used to test the profiling process given software usage data in the frequency domain. Usage data in the duration domain consists of data reflecting how long users are using the software features and interacting with certain use cases. Usage data in the frequency domain reflects how often users invoke certain features and use cases. In the next sections, we will illustrate the user simulation logic in each domain.

The simulation script is programmed in Python 3 with the Selenium plugin. Selenium is a web testing toolkit that allows you to create self-running scripts that interact with web controls.

5.3.2 User Simulation for the Duration Domain

In this scenario, each simulated user is allowed to visit the website only once. We specify four different user types. Each type has a unique input/behaviour pattern. Even though these patterns do not cover all possible user behaviours, these patterns still generate enough data samples to test the prototype. The user types we use in the simulation are listed below.

1. Bounce type: simulated users that visit the homepage only then quit.
2. Basic type: simulated users view blogs and are interested in the author. They might view the contact page.
3. Curious type: simulated users are interested in the author after they view the homepage. They continue to visit the about page to know more about the blog.
4. Core users: simulated users who view blogs, read comments. They also visit the contact and about pages.

When the simulation process begins, the program randomly selects a behaviour type. For each behaviour type, there is an action class that generates random user input. The amount of time spent in each session is randomized. The duration has three levels: 'long', 'medium' and 'short'. A 'long' duration lasts a maximum of 20s, a 'medium' duration lasts a maximum of 12s, and a 'short' duration lasts a maximum of 8s.

The simulation process is repeated 2,000 times. The scenario is simulated that 2,000 unique users visit the website one after the other.

5.3.3 User Simulation for the Frequency Domain

In this scenario, each simulated user visits the website several times. Depending on his usage patterns, he accesses different functions numerous times. As with the usage patterns of the continuous range test, the users in the frequency range test have three unique behaviour types. These types are defined by the frequency of function use.

We program a new input simulation logic. We create a list of 2000 unique users numbered from 1-2000. These numbers are the unique identifiers of each simulated user. For the purposes of variable control, most simulated users are normal users with an intermediate frequency of activity. A small number of users are novice users with low activity frequency. Another small number of users are heavy users who use the application frequently, so we classify these simulated users as follows.

1. Beginner type, user id 1-400. Users only view the main page once.
2. Normal type, user id 401-1700. Users visit the website once and view blog content once or twice.
3. Frequent type, user id 1701 – 2000. Users visit the website multiple times (4 times maximum). Each time they visit the website, they view blogs multiple times (3 times maximum) and make comments multiple times (5 times maximum). They open the about page and contact page randomly.

When the simulation process starts, we simulate user inputs from the beginning of the user list. The program generates user inputs according to the user id. For the frequent users, the program will be executed multiple times (4 times maximum). The program writes these variables into the database after each cycle.

5.3.4 Data Recording

The software requires usage data capturing features to apply the use case based user profiling process. For websites and web applications, software developers can integrate web analytic tools supporting usage data tracking into their software. We discovered a nice, open-source toolkit that captures website runtime usage data among all types of web analytics tools. It is called Open Web Analytics. It is an open-source web analytic tool that can be easily

implemented in a web application. We didn't deploy OWA in the experiment since our web blog application is hosted on the local machine. The OWA only works with websites and web applications deployed on the Internet. Therefore, the OWA service cannot perform well in our experiment environment. As a solution, we stored simulated user inputs data directly into the database that keeps software usage data.

5.3.5 Usage Data Dataset Overview

The user simulation program generated two sets of user behaviour data. The values in the data sets are numerical. These data sets contain information about how the simulated users interact with the blog system.

5.3.5.1 Duration Domain Usage Data Dataset

The first dataset of duration domain user simulation contains nine columns. The values in this dataset represent the amount of time a simulated user spends using each functionality. An overview of the first ten rows of the dataset can be found in Figure 15. The first column of the dataset is the key value. There are eight index values: 'user id', 'homepage', 'Allblogs', 'blog 1', 'blog2', 'comment 1', 'comment 2', 'contact', and 'about page'. A detailed explanation of these values can be found in Table 7.

Table 7 Duration Domain Dataset Values Explanation

INDEX TITLE	MEANING
<i>HOMEPAGE</i>	the total time a user stays on the homepage
<i>ALLBLOGS</i>	the total time a user stays on the blog overview page
<i>BLOG 1</i>	the total time a user views the blog number 1
<i>BLOG 2</i>	the total time a user views the blog number 2
<i>COMMENT 1</i>	the total time a user spends posting a comment under blog 1
<i>COMMENT 2</i>	the total time a user spends posting a comment under blog 2
<i>CONTACT</i>	the total time a user stays on the contact information page
<i>ABOUT PAGE</i>	the total time a user stays on the about information page

Figure 15 Duration Domain Dataset Overview

	Homepage	Allblogs	Blog1	Blog2	Comment1	Comment2	Contact	About
0	9.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	1.9	6.2	0.0	0.0	0.0	0.0	1.0	0.0
3	6.6	0.0	15.3	0.0	0.0	0.0	0.0	0.0
4	4.0	6.6	0.0	8.8	0.0	8.8	2.7	7.1
5	2.2	0.0	1.4	0.0	0.0	0.0	0.0	0.0
6	6.4	0.0	7.9	0.0	0.0	0.0	6.7	0.0
7	7.3	0.0	0.0	0.0	0.0	0.0	5.0	0.0
8	3.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	4.1	0.0	0.0	0.0	0.0	0.0	3.9	0.0

5.3.5.2 Frequency Domain Usage Data Dataset

The second data set of the frequency domain user simulation contains ten columns. The values in this dataset indicate the frequency with which a simulated user interacts with specific features. Each row represents the record of one visit by a user. An overview of the first ten rows of the data set is shown in Figure 16. The first column of the data frame is the key value. There are nine index values, 'User', 'Homepage', 'Allblogs', 'Blog1', 'Blog2', 'Comment1', 'Comment2', 'Contact' and 'About'. Detailed explanations can be found in Table 8.

Table 8 Frequency Domain Dataset Values Explanation

INDEX TITLE	MEANING
<i>USER</i>	the unique id of a user
<i>HOMEPAGE</i>	the number of times a user opens the homepage
<i>ALLBLOGS</i>	the number of times a user opens the blog overview page
<i>BLOG 1</i>	the number of times a user opens the blog number 1
<i>BLOG 2</i>	the number of times a user opens the blog number 2
<i>COMMENT 1</i>	the number of times a user posts a comment under blog 1
<i>COMMENT 2</i>	the number of times a user posts a comment under blog 2
<i>CONTACT</i>	the number of times a user opens the contact information page
<i>ABOUT PAGE</i>	the number of times a user opens the about information page

Figure 16 Frequency Domain Dataset Overview

	User	Homepage	Allblogs	Blog1	Blog2	Comment1	Comment2	Contact	About
0	1	1	0	0	0	0	0	2	2
1	2	1	0	0	0	0	0	1	2
2	3	1	0	0	0	0	0	1	1
3	4	1	0	0	0	0	0	2	1
4	5	1	0	0	0	0	0	2	2
5	6	1	0	0	0	0	0	2	1
6	7	1	0	0	0	0	0	1	1
7	8	1	0	0	0	0	0	1	1
8	9	1	0	0	0	0	0	2	1
9	10	1	0	0	0	0	0	2	1

5.4 Data Aggregation Method

The user profile is created based on the system use case model. Since we focus on profiling software end-users, we map the data to the use cases relevant to the end-users. Figure 17 is the system use case diagram created for use case based user profiling purposes.

This diagram is different from the system use case diagram in figure 14. We only focus on the use cases linked with the visitor actor since we analyse end-users' behaviours only. The blog application administrator is irrelevant in this situation. There are four primary use cases linked with the visitor actor in this use case diagram. Moreover, it contains use cases that express user interactions' subsets. In this use case diagram, there are four primary use cases: 'Visit Home Page', 'Read Blogs', 'Leave Comments', and 'Read Contact Information'. In addition, 'Read Blogs' and 'Leave Comments' each have two other use cases, namely 'Read Blogs 1', 'Read Blogs2', 'Comment1' and 'Comment2'. Table 9 shows the exact meaning of each use case and the use cases subsets.

Figure 17 Blog System Use Case Diagram

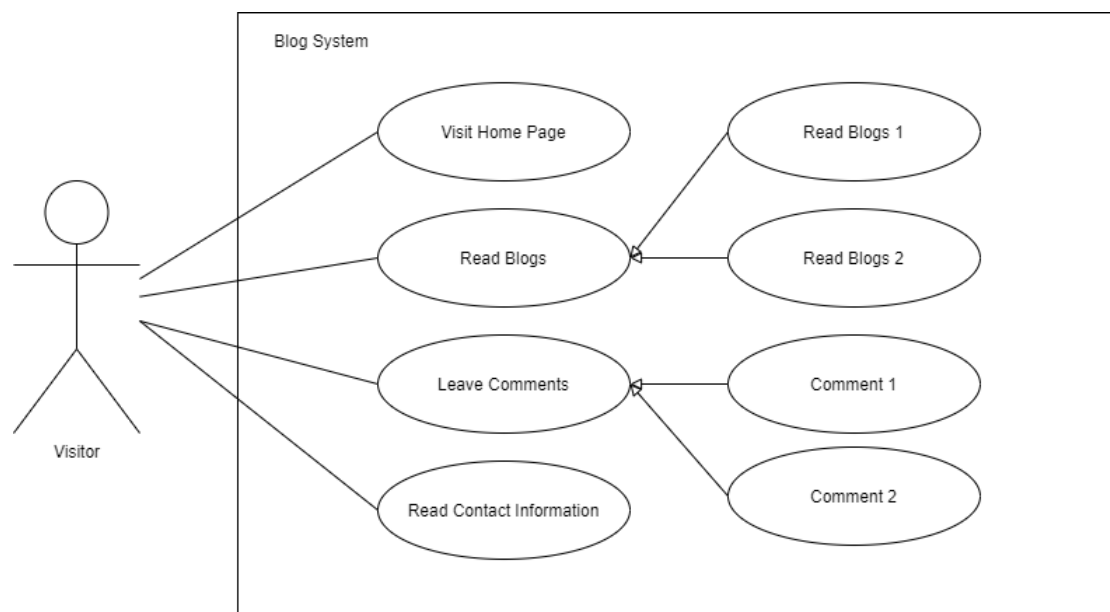


Table 9 Blog System Use Case Explanation

USE CASE TITLE	MEANING
<i>VISIT HOME PAGE</i>	Users view the information on the Homepage and the navigation bar
<i>READ BLOGS</i>	Users read blogs posted on the website
<i>LEAVE COMMENTS</i>	Users leave comments on the blogs
<i>READ CONTACT INFORMATION</i>	Users read information on the contact page and the about page
<i>READ BLOGS 1</i>	Users read blog 1 content
<i>READ BLOGS 2</i>	Users read blog 2 content
<i>COMMENT 1</i>	Users leave comments under blog 1
<i>COMMENT 2</i>	Users leave comments under blog 2

To understand user behaviour based on use cases, we need to rearrange the data. Our method is to map dataset values to use cases. Then we group the values under the same use case. For example, we select the values of 'Comment 1' and 'Comment 2' and add them together. The title of the newly calculated value is 'Comment.' This new value indicates the time spent by users under the 'Leave Comments' use case. The mapping processes are the same for duration and frequency domain analyses. The original dataset has eight index values: 'homepage', 'Allblogs', 'blog 1', 'blog2', 'comment 1', 'comment 2', 'contact', and 'about page'. We group the variables that have the same target use cases. Table 10 is the mapping table:

Table 10 Use Case – Values Mapping Table

ORIGINAL VALUES	TARGET USE CASE
'HOMEPAGE'	Visit Home Page
'ALLBLOGS', 'BLOG 1', 'BLOG 2'	Read Blogs
'COMMENT 1', 'COMMENT 2'	Leave Comments
'CONTACT', 'ABOUT PAGE'	Read Contact Information

Variables representing the usage condition of a single functionality are mapped to the use case subsets. Table 11 is the use case subsets mapping table.

Table 11 Use Case Subsets-Value Mapping Table

ORIGINAL VALUES	TARGET USE CASE
'BLOG 1'	Read Blogs 1
'BLOG 2'	Read Blogs 2
'COMMENT 1'	Read Comment 1
'COMMENT 2'	Read Comment 2

5.5 Use Case Based User Cluster Analyses Results

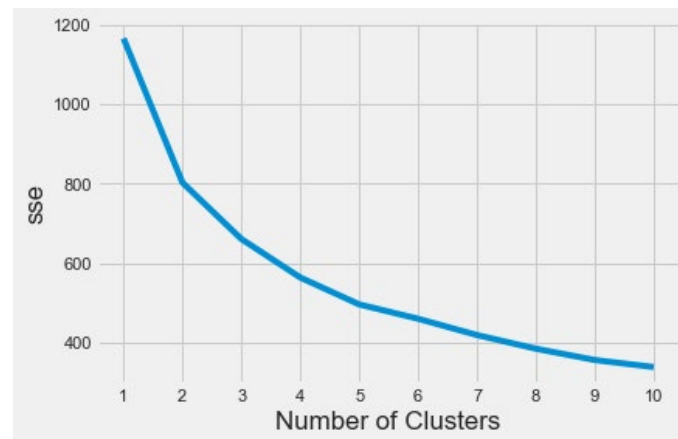
The results of the cluster analyses are presented below. For the cluster analyses using the K-Means algorithm, we present the elbow method table for determining cluster numbers, centroid information, and cluster distribution. For the cluster analyses by agglomerative hierarchical clustering, we present a dendrogram used to determine the cluster number and the distribution of the clusters. We create a heat map in each scenario.

5.5.1 Duration Domain - K-means Clustering

5.5.1.1 Elbow Method

Figure 18 shows the curve where the SSE is a function of the number of clusters. The elbow method indicates that the appropriate number of clusters is 4. We set the cluster numbers to 4 in the k-means clustering process.

Figure 18 Duration Domain SSE as the Function of Number of Clusters



5.5.1.2 K-means Clustering Results

For the software usage data in each cluster, we calculate their mean. We consider several important key attributes that help to understand the state of software usage. These are the number of users in the clusters, the share of each cluster, the mean of the total time a user spends on each visit, and the mean of the time spent on each use case. The values are listed in Table 12. The time values are measured in seconds.

Table 12 Duration Domain K-means Clustering Cluster Overview

	<i>CLUSTER 1</i>	<i>CLUSTER 2</i>	<i>CLUSTER 3</i>	<i>CLUSTER 4</i>
<i>COUNTS</i>	1001	219	203	577
<i>PROPORTION</i>	50.5%	10.95%	10.15%	28.85%
<i>AVG TOTAL TIME(S)</i>	10.82	58.56	54.44	15.99
<i>AVG HOMEPAGE TIME(S)</i>	6.62	6.79	5.99	6.39
<i>AVG BLOG TIME(S)</i>	3.26	27.08	24.84	3.62
<i>AVG COMMENT TIME(S)</i>	0.41	19.97	18.35	0.47
<i>AVG CONTACT TIME(S)</i>	0.43	2.37	2.48	5.39
<i>AVG ABOUT TIME(S)</i>	0.10	2.36	2.78	0.12

5.5.1.3 Use Case Based User Profile

After aggregation, we have a user profile dataset with eight indexes. The first ten elements in the user profile dataset are shown in Table 13. The numbers indicate the duration a user spends under a certain use case.

Table 13 Duration Domain Use Case Based User Profiles

	VISITHOMEPAGE	BLOGS	COMMENT	OTHER	BLOG1	BLOG2	COMMENT1	COMMENT2
0	9.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	1.9	6.2	0.0	1.0	0.0	0.0	0.0	0.0
3	6.6	15.3	0.0	0.0	15.3	0.0	0.0	0.0
4	4.0	15.4	8.8	9.8	0.0	8.8	0.0	8.8
5	2.2	1.4	0.0	0.0	1.4	0.0	0.0	0.0
6	6.4	7.9	0.0	6.7	7.9	0.0	0.0	0.0
7	7.3	0.0	0.0	5.0	0.0	0.0	0.0	0.0
8	3.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	4.1	0.0	0.0	3.9	0.0	0.0	0.0	0.0

5.5.1.4 Heatmap

We use the min-max approach to normalize each column of the dataset. For each cluster, we calculate their use case usage degree mean values. The result is shown in Table 14.

We make a heat map accordingly. Red stands for a high value, and green stands for a low value. The colour indicates how intensively users of each cluster are acting under these use cases. Red implies that they spend a lot of time under the use case on the x-axis. Green means the opposite way. The heatmap is shown in Figure 19.

Table 14 Duration Domain K-means Clustering Clusters Use Case Usage Degree Normalized

	VISITHOME PAGE	BLOGS	COMMENT	OTHER	BLOG1	BLOG2	COMMENT1	COMMENT2
CLUSTER1	0.511080	0.068345	0.010965	0.034511	0.053826	0.070894	0.009985	0.010519
CLUSTER2	0.526069	0.567799	0.533929	0.306618	0.261895	0.736553	0.261895	0.736553
CLUSTER3	0.453292	0.520763	0.490714	0.341373	0.704163	0.213473	0.704163	0.213473
CLUSTER4	0.490074	0.075813	0.012669	0.357796	0.074818	0.062600	0.011404	0.012288

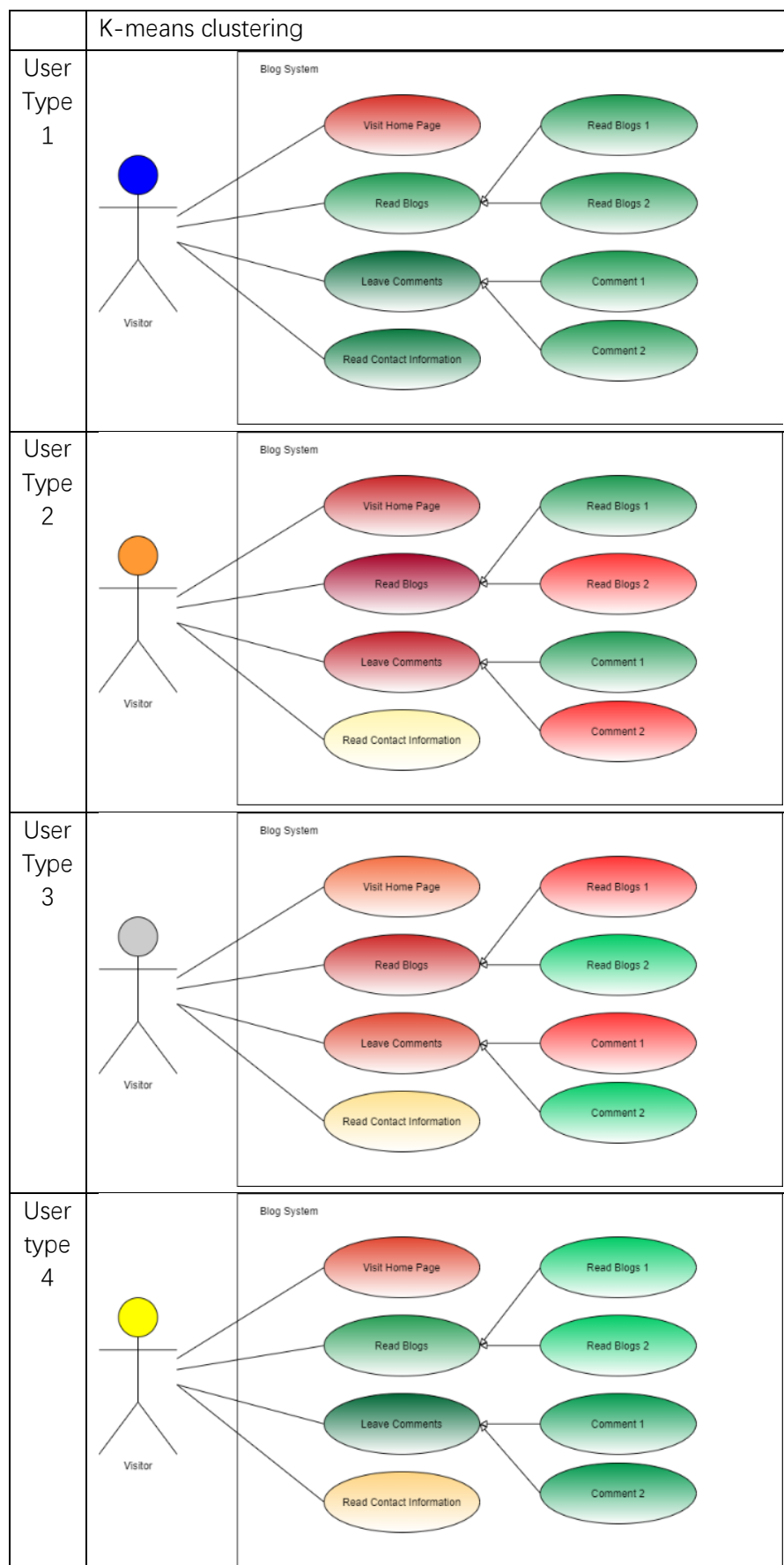
Figure 19 Duration Domain K-means Clustering Use Case Heatmap



5.5.1.5 Infographic Use Case Diagram

The infographic use case diagram representing each user type applies the colours in the heatmap. Each cell in the heatmap is mapped to a use case in the infographic use case diagram. The colours in the infographic use case diagram are the same as those in the heatmap cells of the same use case. User types 1, 2, 3, and 4 are the user groups of clusters 1, 2, 3, and 4. Table 15 shows the infographic use case diagrams of four different user types

Table 15 Duration Domain Infographic Use Case Diagram of K-means Clustering



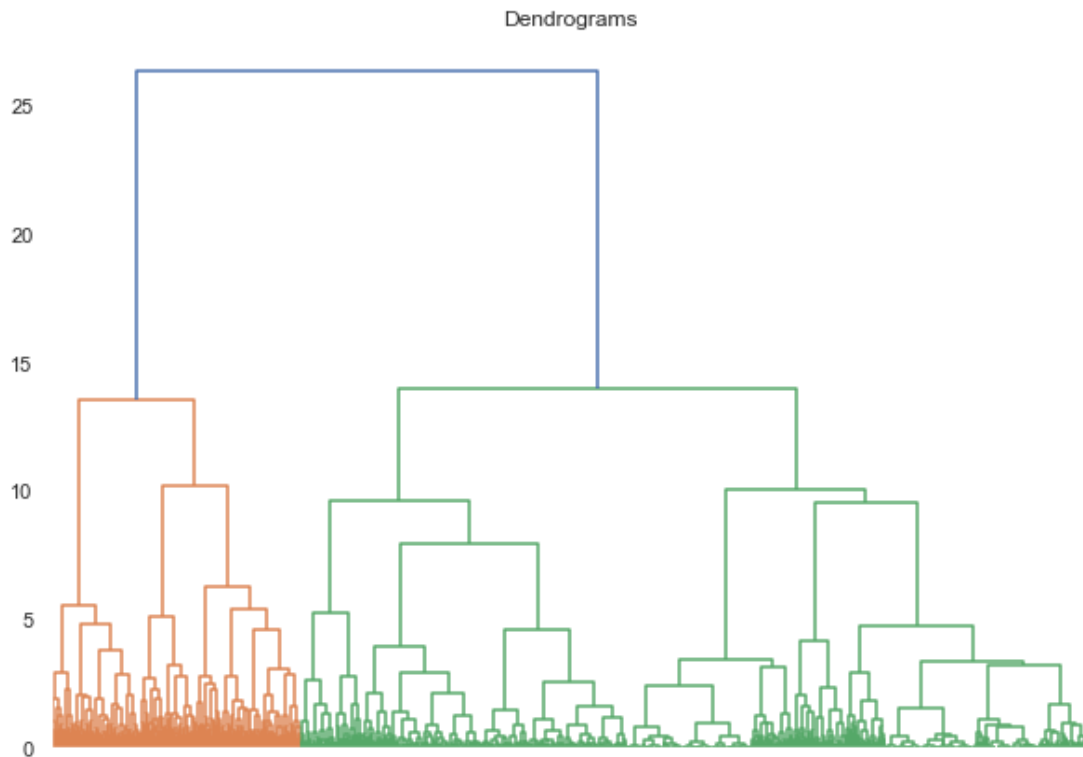
5.5.1.6 Improvement Suggestions

Based on the user profiling results, we can make basic system improvement suggestions. The web blog home page is used most often, and user types 2, 3, 4 visit the home page and contact information simultaneously. We suggest combining the contact information showcase feature with the homepage. It should make it easier for visitors to locate the information they need. We will illustrate the system improvement suggestion in detail in the result comparison sections.

5.5.2 Duration Domain - Hierarchical Clustering

5.5.2.1 Dendrogram

Figure 20 Duration Domain Hierarchical Clustering Dendrogram



Dendrogram from the hierarchical clustering shows how clusters are formed from the bottom. The dendrogram for the duration domain agglomerative hierarchical clustering is shown in Figure 20. To determine the cluster number, we set the number of clusters to 4 according to the dendrogram.

5.5.2.2 Agglomerative Hierarchical Clustering Results

We select some important key attributes that help people understand the state of software usage. We compute their mean value and present them in Table 16. We compute the number of users included in each cluster, the proportion of each cluster, the mean of the total time a user spends on each visit, and the mean of the time spent on each use case. The time values in the table are measured in seconds.

Table 16 Duration Domain Hierarchical Clustering Clusters Overview

	<i>CLUSTER 1</i>	<i>CLUSTER 2</i>	<i>CLUSTER 3</i>	<i>CLUSTER 4</i>
<i>COUNTS</i>	890	310	172	628
<i>PROPORTION</i>	44.5%	15.5%	8.6%	31.4%
<i>AVG TOTAL TIME(S)</i>	10.79	57.02	46.50	13.78
<i>AVG HOMEPAGE TIME(S)</i>	6.85	6.63	6.33	6.01
<i>AVG BLOG TIME(S)</i>	3.02	25.95	20.86	3.19
<i>AVG COMMENT TIME(S)</i>	0.22	19.12	15.06	0.10
<i>AVG CONTACT TIME(S)</i>	0.60	2.79	2.04	4.48
<i>AVG ABOUT TIME(S)</i>	0.10	2.54	2.20	0.002

5.5.2.3 Use Case Based User Profile

The process of creating an application use case based user profile is the same as for K-means clustering. The profile data set is also the same. The only difference is that due to a different clustering algorithm, the composition of each cluster is different. Consequently, the mean value of the time spent on each use case in each cluster may be different.

5.5.2.4 Heatmap

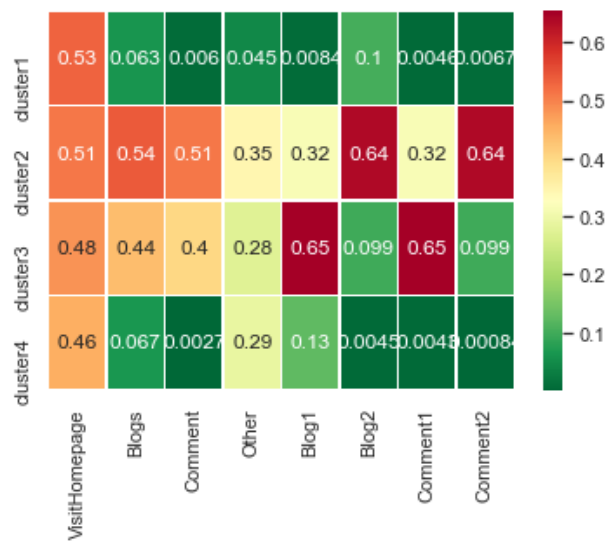
As we have done in k-means clustering, we are using the min-max approach to normalize each column of the dataset. For each cluster, we calculate the mean value of use case usage degrees. The result is shown in Table 17.

Table 17. Duration Domain Hierarchical Clustering Clusters Use Case Usage Degree Normalized

	<i>VISITHOMEPAGE</i>	<i>BLOGS</i>	<i>COMMENT</i>	<i>OTHER</i>	<i>BLOG1</i>	<i>BLOG2</i>	<i>COMMENT1</i>	<i>COMMENT2</i>
CLUSTER1	0.532033	0.063338	0.006009	0.045046	0.008371	0.104612	0.004579	0.006657
CLUSTER2	0.511789	0.544018	0.511247	0.345580	0.315081	0.640952	0.315081	0.640952
CLUSTER3	0.484619	0.437302	0.402780	0.275785	0.653721	0.099477	0.653721	0.099477
CLUSTER4	0.455530	0.066859	0.002738	0.291060	0.127046	0.004482	0.004275	0.000844

We create a corresponding heat map. The colour indicates how intensively the users of each cluster act under these use cases on the x-axis. The heatmap is shown in Figure 21.

Figure 21 Duration Domain Hierarchical Clustering Use Case Heatmap



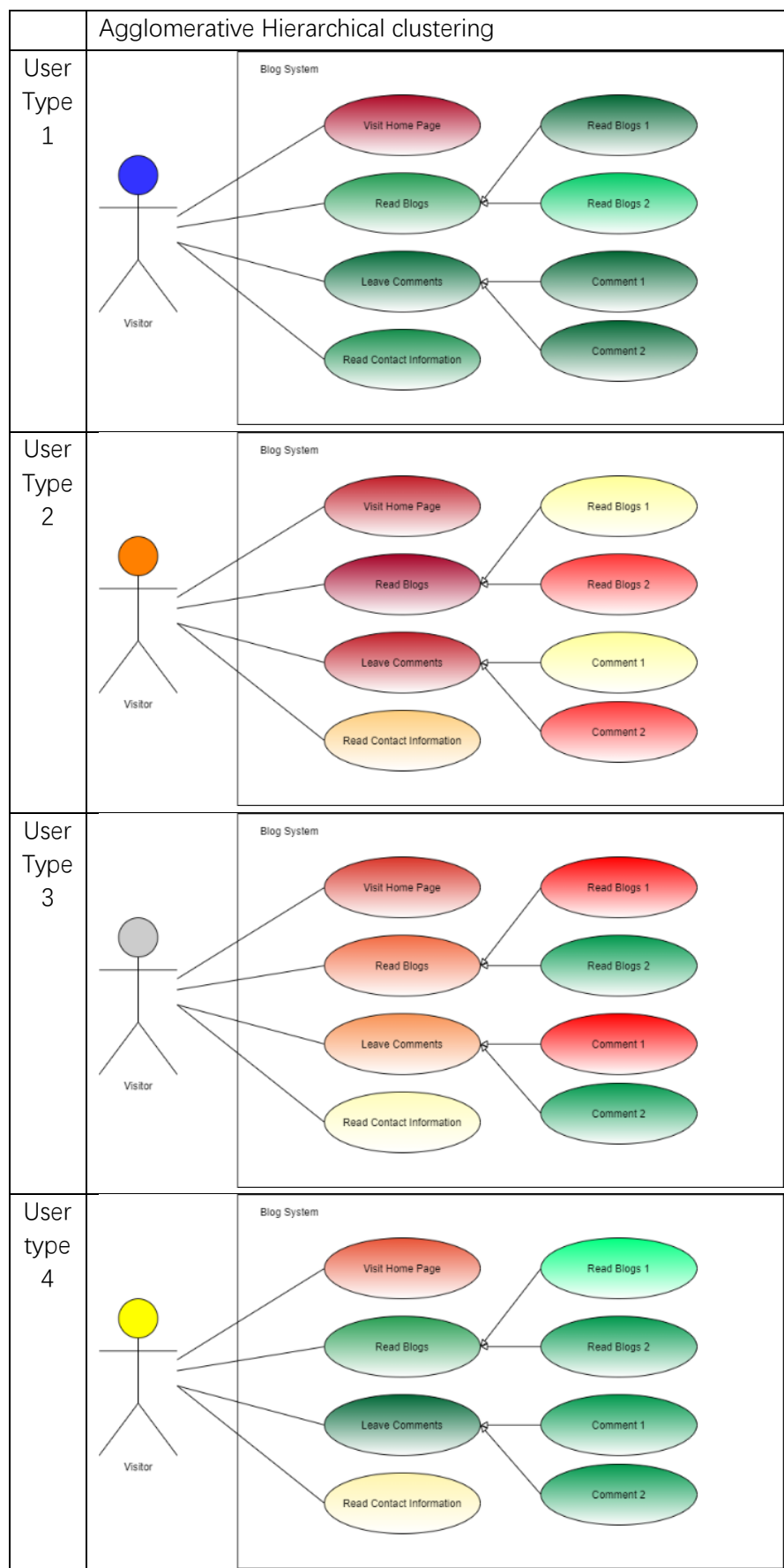
5.5.2.5 Infographic Use Case Diagram

We use the same method of creating an infographic use case diagram with profiling results as the previous section. Table 18 shows the infographic use case diagrams of four different user types. The user type 1, 2, 3, 4 refers to the cluster 1, 2, 3, 4.

5.5.2.6 Improvement Suggestions

Since the user profiling result is similar to that of the K-means clustering, we suggest system improvement by integrating the contact information with the homepage. More details will be discussed in the result comparison section.

Table 18 Duration Domain Infographic Use Case Diagram of Hierarchical Clustering

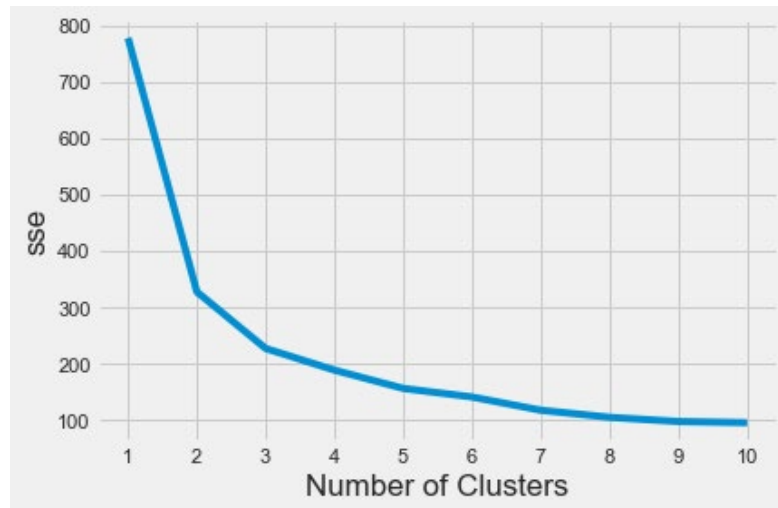


5.5.3 Frequency Domain - K-means Clustering

5.5.3.1 Elbow Method

Figure 22 shows the curve in which SSE is the function of the number of clusters. The elbow method shows that the appropriate number of clusters is 3. We set the number of clusters in the k-means clustering process to 3.

Figure 22 Frequency Domain SSE as the Function of Number of Clusters



5.5.3.2 K-means Clustering Results

In K-means clustering in the frequency domain, software usage data are frequency values. These numbers indicate how often users visit or use certain features. For the software frequency data in each cluster, we calculate its mean value. We select some key attributes for demonstration. These are the number of users included in the cluster, the frequency with which a user visits the website, and the mean of the frequency with which a user accesses each feature. The values are shown in Table 19.

Table 19 Frequency Domain K-means Clustering Cluster Overview

	CLUSTER1	CLUSTER2	CLUSTER3
COUNTS	437	1355	208
FREQUENT	1	1	3.56
HOMEPAGE	1	1	2.56
ALLBLOGS	0.08	1.08	5.02
BLOG1	0.08	2.03	5.07
BLOG2	0.08	2.05	5.11
COMMENT1	0	0.22	7.65
COMMENT2	0	0.19	7.73
ABOUT	1.44	0.49	1.32
CONTACT	1.42	0.49	1.36

5.5.3.3 Use Case Based User Profile

After aggregation, we have a user profile dataset with nine indexes. The first ten elements of the user profile dataset in the frequency domain are listed in Table 20. The numbers in the table indicate how often users access features under specific use cases.

Table 20 Frequency Domain Use Case Based User Profiles

	VISITHOMEPAGE	BLOGS	COMMENT	OTHER	BLOG1	BLOG2	COMMENT1	COMMENT2
0	1	0	0	4	0	0	0	0
1	1	0	0	3	0	0	0	0
2	1	0	0	2	0	0	0	0
3	1	0	0	3	0	0	0	0
4	1	0	0	4	0	0	0	0
5	1	0	0	3	0	0	0	0
6	1	0	0	2	0	0	0	0
7	1	0	0	2	0	0	0	0
8	1	0	0	3	0	0	0	0
9	1	0	0	3	0	0	0	0

5.5.3.4 Heatmap

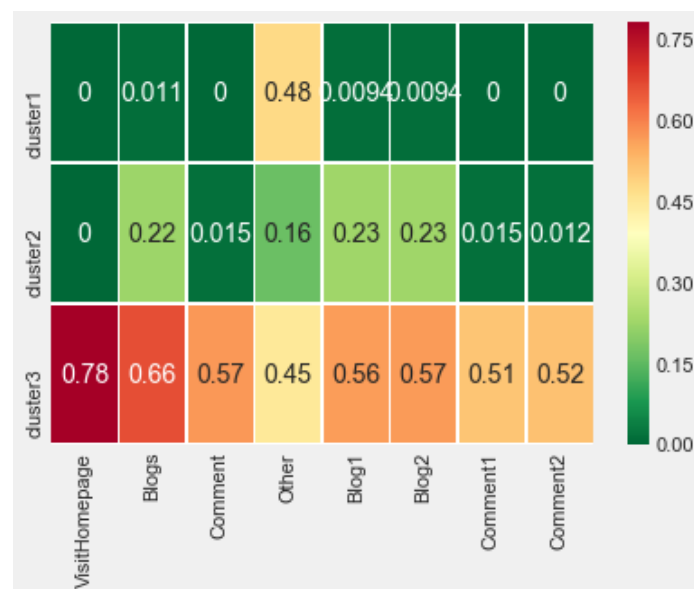
We use the min-max approach to normalize each column of the dataset. For each cluster, we calculate the mean value of use case usage degrees. The result is shown in table 21.

Table 21. Frequency Domain K-means Clustering Clusters Use Case Usage Degree Normalized

	VISITHOMEPAGE	BLOGS	COMMENT	OTHER	BLOG1	BLOG2	COMMENT1	COMMENT2
CLUSTER1	0.000000	0.011044	0.000000	0.475973	0.009408	0.009408	0.000000	0.000000
CLUSTER2	0.000000	0.224001	0.015006	0.162116	0.225748	0.227224	0.014662	0.012349
CLUSTER3	0.778846	0.660744	0.569444	0.446314	0.563034	0.567308	0.509936	0.515064

And we create a corresponding heat map. Red represents high frequency, and the green represents low frequency. The colour indicates how frequently the users of each cluster use the functions under the use cases on the x-axis. Red means that the corresponding use case is frequently invoked by users. Green means that the opposite is true. The heatmap is shown in Figure 23.

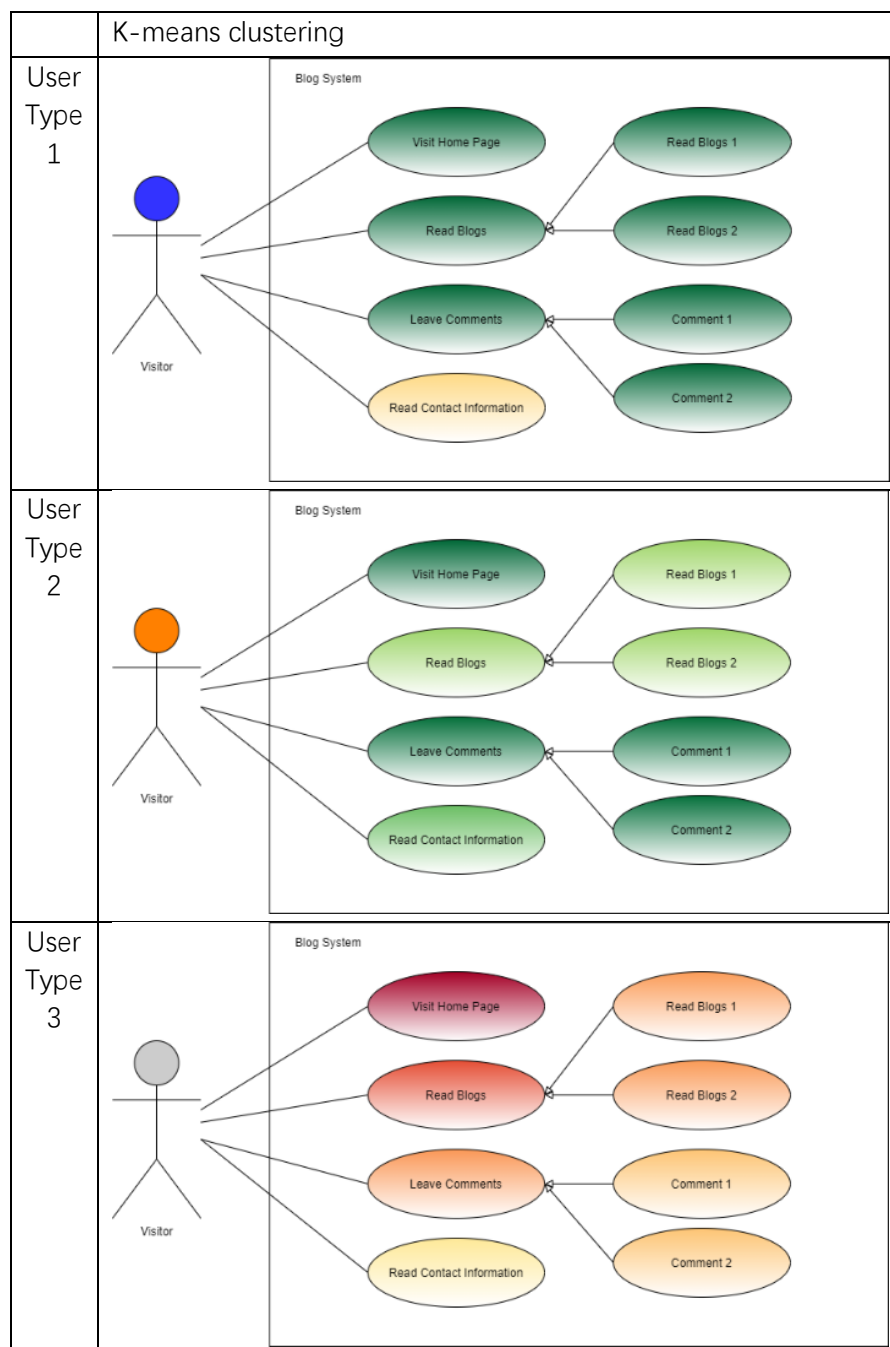
Figure 23 Frequency Domain K-means Clustering Use Case Heatmap



5.5.3.5 Infographic Use Case Diagram

Table 22 shows the infographic use case diagrams of three different user types. The user type 1, 2, 3 refers to the cluster 1, 2, 3. The colours of the use cases are the same as the cells in the heatmap.

Table 22 Frequency Domain Infographic Use Case Diagram of K-means Clustering



5.5.3.6 Improvement Suggestions

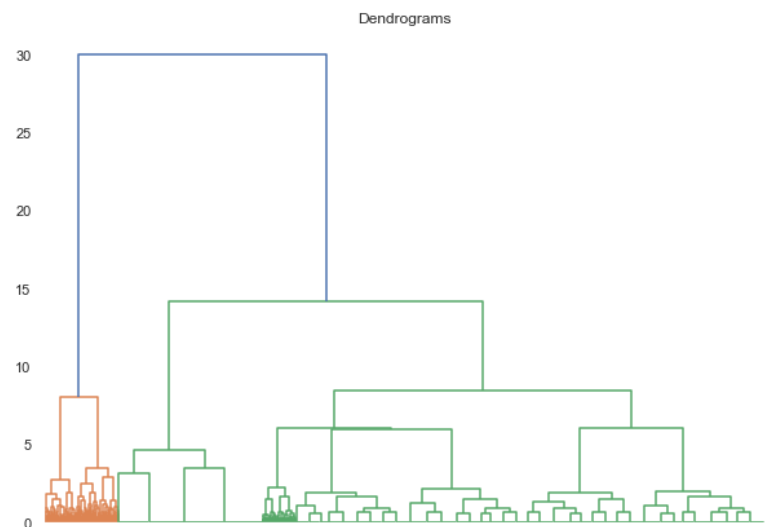
In the user profiling result in the frequency domain with K-means Clustering, there are three user types. They are differentiated in how often users invoke certain use cases and web blog features. User type 3 has a user profile with many red colours in all use cases, which means type 3 users visit all features very often. On the contrary, type 1 and type 2 users seldom leave any comments. We suggest integrating showcase the blog content and contact

information together to the homepage, such that visitors of type 1 and 2 can follow news feeds easily by browsing one page. Frequent users of type 3 may click on the blog content page to leave comments. More details will be discussed in the result comparison section.

5.5.4 Frequency Domain - Hierarchical Clustering

5.5.4.1 Dendrogram

Figure 24 Frequency Domain Hierarchical Clustering Dendrogram



The dendrogram is created for determining the cluster numbers. In this situation, we choose to set the number to three. The dendrogram is shown in Figure 24.

5.5.4.2 Agglomerative Hierarchical Clustering Results

Similar to K-means clustering in the frequency domain, we select some key attributes for demonstration and calculate their mean values in each cluster. The attributes we choose are the number of users included in the cluster, the frequency with which a user visits the website, and the mean of the number of times a user accesses each feature. The values are listed in Table 23. The values in the first row, 'counts', are the total number of items included in the cluster. The other values in the table are the average frequency with which users activate certain functionalities.

Table 23 Frequency Domain Hierarchical Clustering Clusters Overview

	CLUSTER1	CLUSTER2	CLUSTER3
COUNTS	400	1392	208
FREQUENT	1	1.07	3.56
HOMEPAGE	1	1	2.56
ALLBLOGS	0	1.07	5.02
BLOG1	0	2	5.07
BLOG2	0	2	5.11
COMMENT1	0	0.21	7.65
COMMENT2	0	0.18	7.72
ABOUT	1.48	0.5	1.32
CONTACT	1.46	0.5	1.36

5.5.4.3 Use Case Based User Profile

The user profile dataset is the same as that of the K-means clustering. However, the composition of each cluster is different from that of the K-means clustering. Consequently, the mean values of the number of times accessing each use case in each cluster may be different.

5.5.4.4 Heatmap

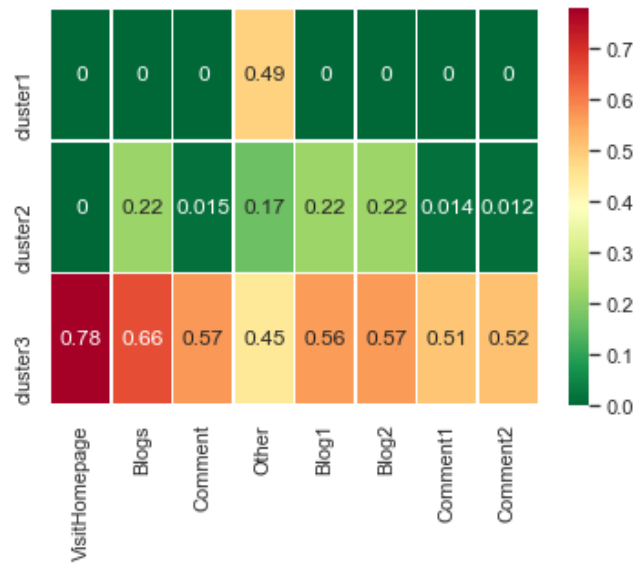
We use the min-max approach to normalize each column of the dataset. Then, for each cluster, we calculate the mean value of use case usage degrees. The result is shown in Table 24.

Table 24 Frequency Domain Hierarchical Clustering Clusters Use Case Usage Degree Normalized

	VISITHOME PAGE	BLOGS	COMMENT	OTHER	BLOG1	BLOG2	COMMENT1	COMMENT2
CLUSTER1	0.000000	0.000000	0.000000	0.489167	0.000000	0.000000	0.000000	0.000000
CLUSTER2	0.000000	0.221514	0.014607	0.166667	0.222701	0.224138	0.014272	0.012021
CLUSTER3	0.778846	0.660744	0.569444	0.446314	0.563034	0.567308	0.509936	0.515064

We make a heatmap accordingly. Figure 25 is the heatmap of the Hierarchical clustering result in the frequency domain. The colour indicates how frequently users of each cluster are accessing functionalities under certain use cases on the x-axis.

Figure 25 Frequency Domain Hierarchical Clustering Use Case Heatmap



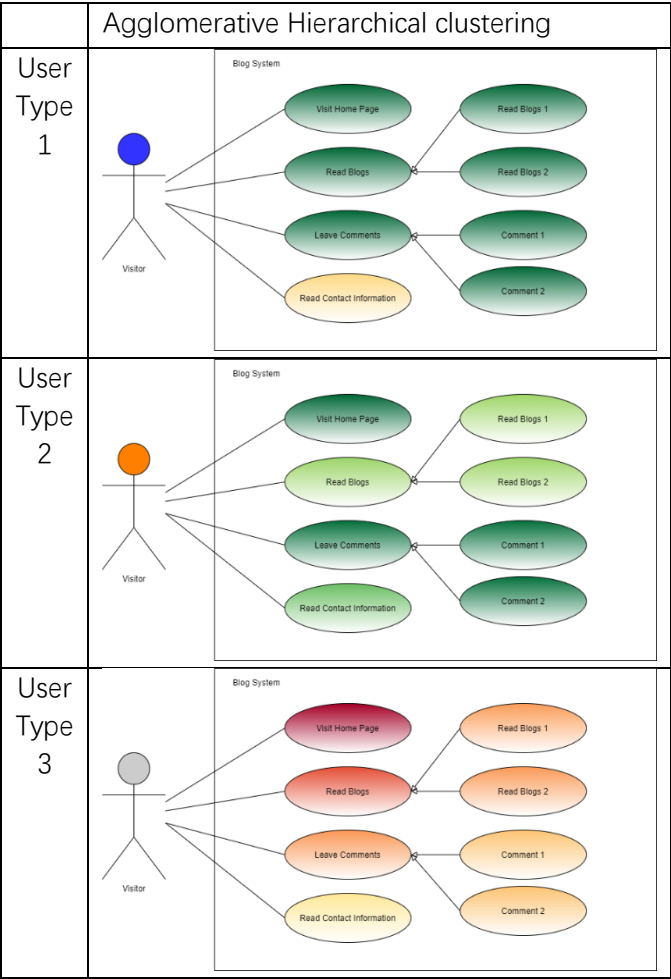
5.5.4.5 Infographic Use Case Diagram

Table 25 is the infographic use case diagrams of three different user types. In the table, the user type 1, 2, 3 refers to cluster 1, 2, 3.

5.5.4.6 Improvement Suggestions

Since the user profiling results are similar to that of the K-means clustering in the frequency domain, user type 3 frequently invokes every feature and use case. Type 1 and type 2 users don't leave any content. We suggest displaying the blog content and contact information on the homepage. We will discuss the system improvement suggestions in the result comparison section.

Table 25 Frequency Domain Infographic Use Case Diagram of Hierarchical Clustering

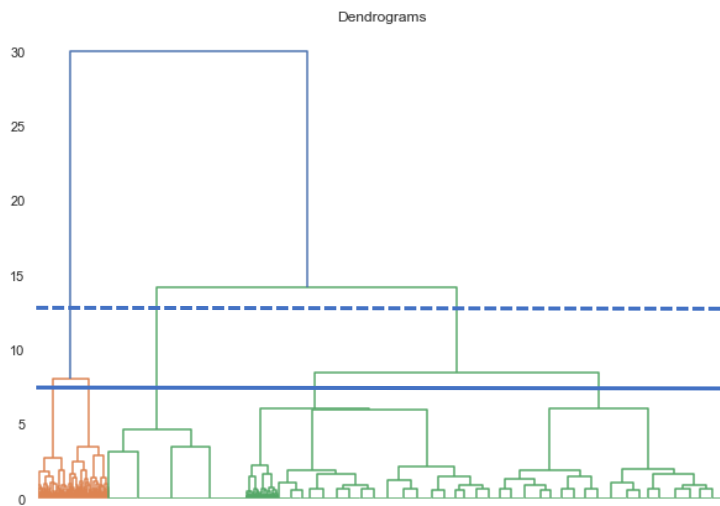


5.5.5 Frequency Domain - Hierarchical Clustering for User Subtypes

This agglomerative hierarchical cluster analysis is used to demonstrate the 'Subtype Discovery' feature of hierarchical clustering. In this cluster analysis, the same user profile dataset is used as in the last hierarchical cluster analysis. However, the number of clusters is set to five. The reason for this is explained as follows.

5.5.5.1 Subtype Discovery with Dendrogram

Figure 26 Dendrogram of the Subsequent User Type Discovery



The dotted line on the dendrogram is the original horizontal line created for cluster numbers choosing. The line below is the new line made for subtype discovery.

Using the dendrogram, we can see how small clusters form larger clusters. In the last cluster analysis, we set the cluster numbers to three. This is because we draw a horizontal line that crosses three vertical lines that represent the differences between the clusters. If we move the line down, we can see that the horizontal line crosses five vertical lines. These are the sub-clusters that make up the last three larger clusters. So by analysing these five clusters, we can see how the subtype user groups form the larger user group. We start the agglomerative hierarchical clustering algorithm again, but with a value of five clusters. The dendrogram is shown in Figure 26.

5.5.5.2 Agglomerative Hierarchical Clustering Results

The result is shown in table 26. To compare it with the result of the last hierarchical cluster analysis, we use the same index. However, in this table, there are five clusters. As in the previous table of hierarchical clustering results, the values in this table represent the average number of activations of certain functions by users, except for the row of 'counts,' which indicates the number of items included in each cluster.

Table 26 Frequency Domain Hierarchical Clustering for Subtypes Clusters Overview

	CLUSTER1	CLUSTER2A	CLUSTER2B	CLUSTER3A	CLUSTER3B
COUNTS	400	657	735	116	92
FREQUENT	1	1	1.13	4	3
HOME PAGE	1	1	1	3	2
ALL BLOGS	0	1	1.14	5.81	4.03
BLOG1	0	1.99	2.02	5.91	4
BLOG2	0	1.997	2.04	6.02	3.96
COMMENT1	0	0	0.41	8.81	6.18
COMMENT2	0	0	0.34	9.13	5.96
ABOUT	1.48	0.49	0.51	1.51	1.09
CONTACT	1.46	0	0.95	1.62	1.02

From the values in the 'counts' attribute, we can identify that cluster 2A and cluster 2B form cluster 2 in the last hierarchical cluster analysis, cluster 3A and cluster 3B form cluster 3 in the last analysis.

5.5.5.3 Use Case Based User Profile

In this situation, we have five user types deriving from the previous ones. We will calculate mean values for these subtypes.

5.5.5.4 Heatmap

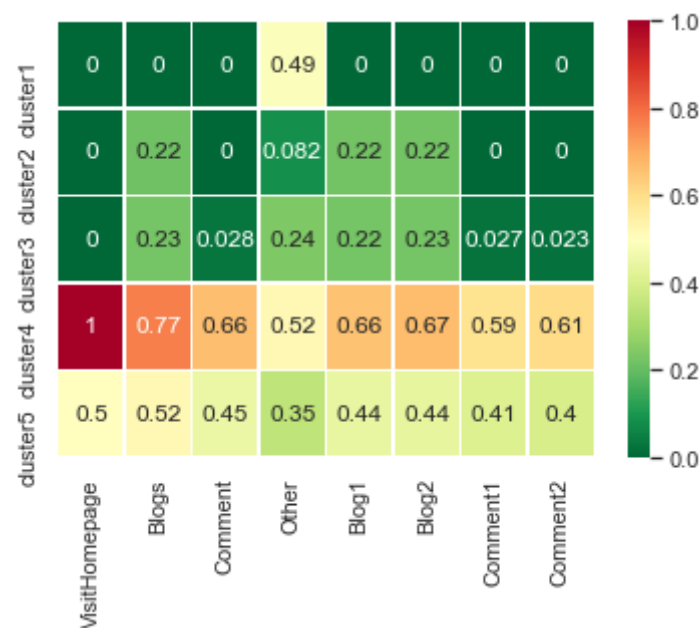
We use the min-max approach to normalize each column of the dataset. For each cluster, we calculate the mean value of use case usage degrees. The results are shown in Table 27.

Figure 27 is a heatmap of the subtypes discovery user profiling analysis. The colour indicates how frequently users of each cluster are accessing functionalities under certain use cases on the x-axis.

Table 27 Frequency Domain Hierarchical Clustering for Subtypes Clusters Use Case Usage Degree Normalized

	VISITHOMEPAGE	BLOGS	COMMENT	OTHER	BLOG1	BLOG2	COMMENT1	COMMENT2
CLUSTER1	0.0	0.000000	0.000000	0.489167	0.000000	0.000000	0.000000	0.000000
CLUSTER2	0.0	0.216730	0.000000	0.082192	0.220869	0.221884	0.000000	0.000000
CLUSTER3	0.0	0.225791	0.027664	0.242177	0.224339	0.226153	0.027029	0.022766
CLUSTER4	1.0	0.771364	0.664432	0.521552	0.657088	0.668582	0.587356	0.608621
CLUSTER5	0.5	0.521267	0.449678	0.351449	0.444444	0.439614	0.412319	0.397101

Figure 27 Frequency Domain Hierarchical Clustering for Subtypes Use Case Heatmap



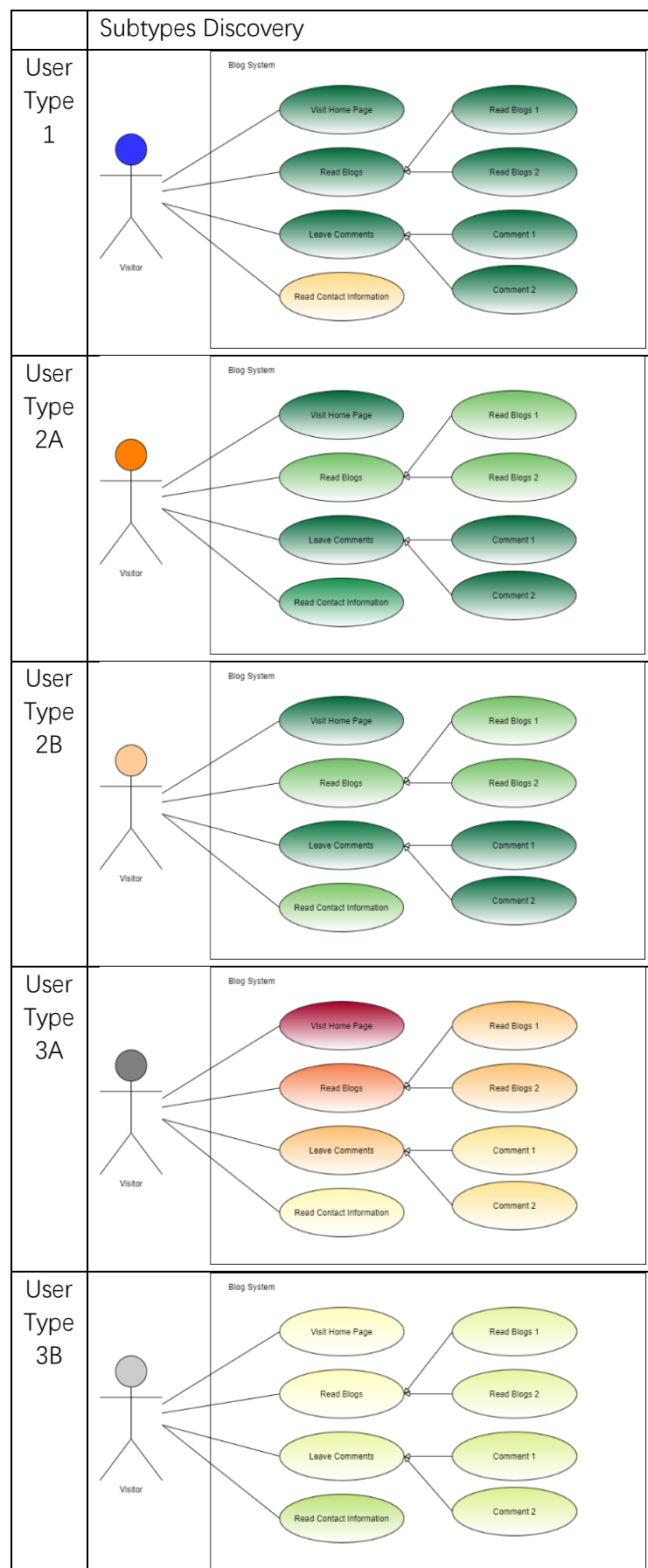
5.5.5.5 Infographic Use Case Diagram

Table 28 shows the infographic use case diagrams of subtypes. The user types 2A, 2B, 3A, 3B are the subtypes of the user type in the previous cluster analysis. User type 1 is the basic type. The subtypes discovery cluster analysis didn't find any user subtypes in user type 1

5.5.5.6 Improvement Suggestions

Subtypes 2A, 2B are similar to their parent user type in user behaviours, so are subtypes 3A, 3B. Thus, the improvement suggestion is the same as we have drawn in the frequency domain hierarchical clustering analysis. The blog content and contact information can be shown on the homepage in the next software update.

Table 28 Frequency Domain Infographic Use Case Diagram of Subtype Discovery



5.6 Result Comparison in the Duration Domain

In the previous sections, we provided an overview of the cluster analyses results. However, listing the statistical data of clusters in tables does not tell us anything about which cluster analysis method performs better in identifying groups of users. To address this issue, we will batch compare the cluster analysis results generated by the same user simulation.

The objectives of the result comparison in this chapter are listed as follows:

1. Visualizing the user type distribution condition
2. Visualizing software usage conditions by using case diagrams
3. Explaining the differences between user types with use case diagrams
4. Discussing how the K-means algorithm and agglomerative hierarchical algorithm affects the user profiling results

Consequently, the results of cluster analysis from K-means clustering and agglomerative hierarchical clustering in duration are discussed together. The results of cluster analysis by K-means clustering, hierarchical clustering, and hierarchical clustering for subtypes in the frequency domain are discussed. In the end, we conclude how two algorithms perform in this user profiling scenario and which one of them performs better overall

User simulation in the Duration section generates usage data representing the duration of users' activities. We use the K-means and agglomerative hierarchical algorithms to classify users into heterogeneous user groups. We set the number of clusters in both cluster analyses to four.

In the results, each cluster is considered as a collection of users. We define that users within the same group belong to one user type. Thus, each cluster analysis in the Duration domain generates four user types. To make the comparison more comprehensive, we name the user types with similar usage behaviour between two cluster results in the same way.

5.6.1 Clusters Overview

This section compares the differences in characteristics of clusters in the duration domain clustering analyses results. Users in the same cluster are regarded as the same user type. For example, users in cluster 1 are taken as user type 1.

5.6.1.1 Duration Domain K-means Clustering Analysis Clusters Overview

Table 29 indicates that type 1 users are the majority who spend the least time on the website on average. Type 1 Users spend more time on the homepage than the other parts. In addition, type 1 users read blogs for a short duration on average.

Type 2 users and type 3 users are similar. They are close in numbers and behaviour patterns. These two types of users spend much time reading blogs and leaving comments. The differences between these types are their preferred blog contents. Type 2 users prefer reading and commenting on blog 2, while type 3 users prefer blog 1.

Type 4 users spend more time reading the contact page than type 1 users. As a result, they spend around five seconds more on the entire website than type 1 users on average. Type 4 users spend approximately five seconds reading the contact page on average, while type 1 users spend almost zero seconds reading contact information.

Table 29 Duration Domain K-means Clustering Analysis Clusters Overview

	<i>TYPE 1</i>	<i>TYPE 2</i>	<i>TYPE 3</i>	<i>TYPE 4</i>
<i>COUNTS</i>	1001	219	203	577
<i>PROPORTION</i>	50.5%	10.95%	10.15%	28.85%
<i>AVG TOTAL TIME(S)</i>	10.82	58.56	54.44	15.99
<i>AVG HOMEPAGE TIME(S)</i>	6.62	6.79	5.99	6.39
<i>AVG BLOG TIME(S)</i>	3.26	27.08	24.84	3.62
<i>AVG BLOG 1 TIME(S)</i>	1.08	<u>5.24</u>	<u>14.08</u>	1.50
<i>AVG BLOG 2 TIME(S)</i>	1.42	<u>14.73</u>	<u>4.27</u>	1.25
<i>AVG COMMENT TIME(S)</i>	0.41	19.97	18.35	0.47
<i>AVG COMMENT 1 TIME(S)</i>	0.20	<u>5.24</u>	<u>14.08</u>	0.23
<i>AVG COMMENT 2 TIME(S)</i>	0.21	<u>14.73</u>	<u>4.27</u>	0.25
<i>AVG CONTACT TIME(S)</i>	0.43	2.37	2.48	5.39
<i>AVG ABOUT TIME(S)</i>	0.10	2.36	2.78	0.12

5.6.1.2 Duration Domain Hierarchical Clustering Analysis Clusters Overview

Table 30 indicates that type 1 users are the majority who spend the least time on the website on average. Type 1 Users spend more time on the homepage than the other parts.

Type 2 users and type 3 users are different in their preferred blog contents. For example, type 2 users prefer reading and commenting on blog 2, while type 3 users prefer blog 1.

Type 4 users spend approximately three more seconds reading the contact page than type 1

users. As a result, they spend around three seconds more on the entire website than type 1 users on average.

Table 30 Duration Domain Hierarchical Clustering Analysis Clusters Overview

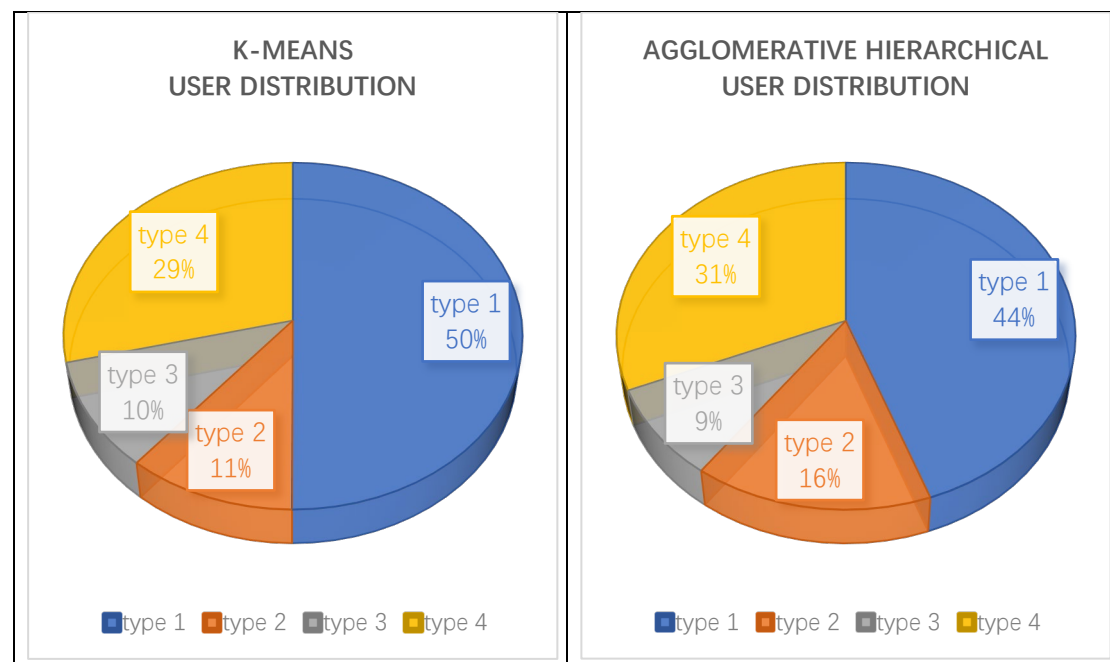
	<i>TYPE 1</i>	<i>TYPE 2</i>	<i>TYPE 3</i>	<i>TYPE 4</i>
<i>COUNTS</i>	890	310	172	628
<i>PROPORTION</i>	44.5%	15.5%	8.6%	31.4%
<i>AVG TOTAL TIME(S)</i>	10.79	57.02	46.50	13.78
<i>AVG HOMEPAGE TIME(S)</i>	6.85	6.63	6.33	6.01
<i>AVG BLOG TIME(S)</i>	3.02	25.95	20.86	3.19
<i>AVG BLOG 1 TIME(S)</i>	0.17	<u>6.30</u>	<u>13.07</u>	2.54
<i>AVG BLOG 2 TIME(S)</i>	2.09	<u>12.82</u>	<u>1.99</u>	0.09
<i>AVG COMMENT TIME(S)</i>	0.22	19.12	15.06	0.10
<i>AVG COMMENT 1 TIME(S)</i>	0.09	<u>6.30</u>	<u>13.07</u>	0.09
<i>AVG COMMENT 2 TIME(S)</i>	0.13	<u>12.82</u>	<u>1.99</u>	0.02
<i>AVG CONTACT TIME(S)</i>	0.60	2.79	2.04	4.48
<i>AVG ABOUT TIME(S)</i>	0.10	2.54	2.20	0.002

5.6.2 User Type Distribution

We create a pie chart for each clustering result to illustrate the difference between the distribution of user types in K-means clustering and hierarchical clustering in the duration domain. The pie charts clearly show the differences in the distribution of user types between the K-means clustering and hierarchical clustering results.

In the K-means clustering result, user type 1 has a share of 50%, while in the hierarchical clustering result, user type 1 has a share of 44%. 11% of the users in the K-means clustering results are user type 2, and 16% of the users in the hierarchical clustering are user type 2. The share of user type 3 and user type 4 in the K-means clustering results is the same as in the hierarchical clustering results.

Figure 28 Duration Domain User Type Distribution Comparison



5.6.3 User Type Comparison by Use Cases

We compare similar user types in the K-means clustering results and the hierarchical clustering results. We calculate the average time of use of the use case for each user type. The data are presented in Table 31. All values are measured in seconds. The capital letter 'K' in the index row represents a user type from the K-means clustering result. The capital letter 'H' in the index field indicates a user type from the hierarchical clustering result. The table shows that the differences between the same user type in the two clustering results are small. The two user types of type 1 of the two clustering results show similar trends in the use of use cases. This is also true for the other user types.

Table 31 Duration Domain User Type Comparison By Use Cases

	TYPE 1 (K)	TYPE 1 (H)	TYPE 2 (K)	TYPE 2 (H)	TYPE 3 (K)	TYPE 3 (H)	TYPE 4 (K)	TYPE 4 (H)
VISIT HOME PAGE	6.62	6.85	6.79	6.63	5.99	6.33	6.39	6.01
READ BLOGS	3.26	3.02	27.08	25.95	24.84	20.86	3.62	3.19
LEAVE COMMENTS	0.41	0.22	19.97	19.12	18.35	15.06	0.47	0.10
READ CONTACT INFO	0.53	0.69	4.72	5.32	5.26	4.25	5.51	4.48
BLOG 1	1.08	0.17	5.24	6.30	14.08	13.07	1.50	2.54
BLOG 2	1.42	2.09	14.73	12.82	4.27	1.99	1.25	0.09
COMMENT 1	0.20	0.09	5.24	6.30	14.08	13.07	0.23	0.09
COMMENT 2	0.21	0.13	14.73	12.82	4.27	1.99	0.25	0.02

5.6.4 *Infographic Use Case Diagrams Comparison*

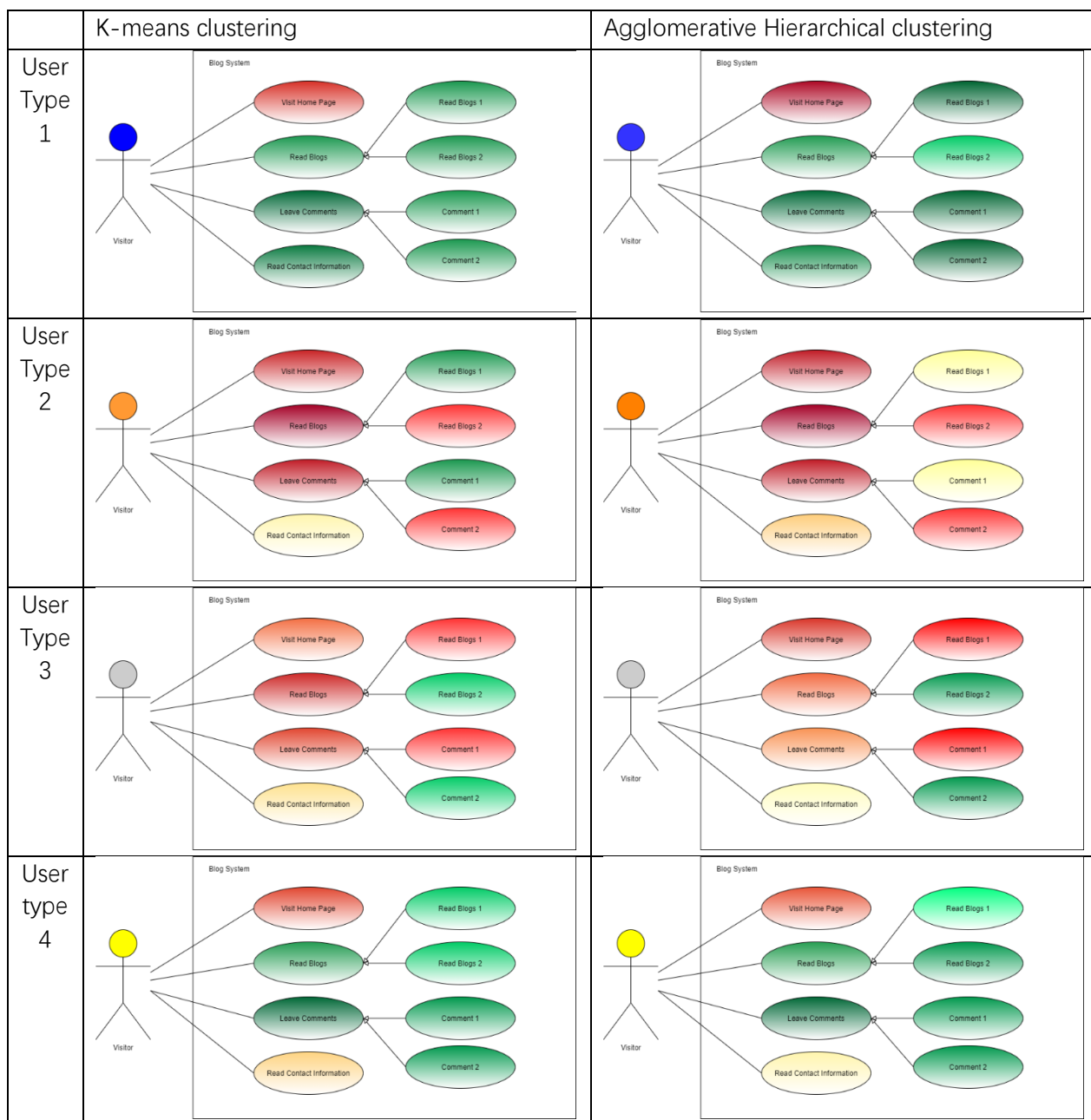
For each user type in the K-means clustering results and the hierarchical clustering results, an infographic use case diagram is created. These use case diagrams have the same use case structure as the blog system use case diagram. We colour the use case eclipses with different colours. We use the colour scheme from the heatmaps created in the clustering results. We use the colours representing the useful life of the use cases in the heatmaps to paint the eclipses in the infographic use case diagram. Thus, the colours in the infographic use case diagrams indicate how long users of a particular user type were active on average under different use cases. The redder the colours within the eclipses, the longer the users use the functionalities under the depicted use cases. On the contrary, the greener the colours within the eclipses, the shorter the users use the functionalities under the presented use cases.

The infographic use case diagrams of four user types from the results of K-means clustering and hierarchical clustering are shown in Table 32. Based on the colour of the eclipses, users can easily identify the usage conditions of the different use cases. For example, it can be seen that both the type 1 users of K-means clustering and hierarchical clustering spend a lot of time on the 'visit homepage' use case. However, they spend very little time on the other use cases because the corresponding eclipses have very green colours.

One can also notice that Type 2 and Type 3 users have similar average usage times for use cases such as 'visit home page', 'read blogs', 'leave comments', and 'read contact information', as the colour of these use case eclipses is very similar. However, for the use case subsets, unlike type 2 users, type 3 users have a red colour within the eclipses of the use cases 'read blog 1' and 'comment 1', which means that type 3 users prefer the first blog to the second blog and are more willing to leave comments in the first blog.

Type 4 users spend more time on the use case 'read contact information' than Type 1 in both clustering results, and the eclipse colour of the use case 'read contact information' is more yellow for Type 4. Yellow represents a longer duration than green. Humans can quickly perceive such a difference in colour.

Table 32 Duration Domain Infographic Use Case Diagram Comparison



5.6.5 Conclusions and Discussions in Duration Domain

With the information we have in the comparison section, we can now make suggestions on how to improve the system. In the meantime, we compare the K-means algorithm and the agglomerative hierarchical algorithm in terms of the quality of the user profiling results.

5.6.5.1 System Improvement Suggestions

In general, both clustering results show that almost half of the blog system users bounce on the homepage. We suggest improving the usability of the homepage and making it more appealing so that more users stay on the site and become interested in the blogs.

Type 1 and Type 4 users make up three-quarters of the total user base. These users spend little time reading blogs or leaving comments. Type 2 and Type 3 are the professional users who read blogs and leave comments. Therefore, we suggest making an adaptive portal page that classifies visitors by their IP addresses. It grants professional users access to read blogs and leave comments. New users may see only basic information on the homepage. Moreover, contact information should be displayed on the home page. This way, new visitors will get all the information they want on the home page and have a better user experience.

5.6.5.2 Clustering Algorithm Comparison

The clustering result produced by the K-means algorithm is similar to that produced by the agglomerative hierarchical algorithm. There are differences in the distribution of user types. There are fewer type 1 users in the hierarchical clustering result. Also, the type 2 users of the hierarchical clustering result spend more time in blog 1 on average than the users of the K-means clustering result. Apart from these aspects, both algorithms generate the same result for the user profile. We conclude that the K-means algorithm and the agglomerative hierarchical algorithm generate identical results in user profiling with usage data in the duration domain when the number of clusters is the same.

5.7 Result Comparison in the Frequency Domain

User simulation in the frequency domain generates usage data representing the frequency of users' activities. K-means clustering and agglomerative hierarchical clustering are applied. In both cluster analyses, we set the number of clusters to three.

The results of K-means clustering and hierarchical clustering yield three instinctive user types. We name the user types with similar usage behaviour between two clustering results in the same way.

5.7.1 Clusters Overview

This section compares the characteristics of clusters in the frequency domain clustering analyses results. Each user type represents one unique cluster.

5.7.1.1 Frequency Domain K-means Clustering Analysis Clusters Overview

Table 33 indicates that type 1 users visit the website only once on average. They rarely read

blog content or leave comments. They may read contact and about information once.

Type 2 users are the majority who read blogs on the website. They visit the website once on average, and they read blogs multiple times. Type 2 users seldom leave comments or visit other parts of the website.

Type 3 users are the frequent users who visit the website multiple times. They visit the webpage 3.56 times on average. In addition, type 3 users read blogs 1 and blog 2 five times on average and leave seven comments on each blog.

Table 33 Frequency Domain K-means Clustering Analysis Clusters Overview

	TYPE 1	TYPE 2	TYPE 3
COUNTS	437	1355	208
FREQUENT	1	1	3.56
HOMEPAGE	1	1	2.56
BLOG1	0.08	2.03	5.07
BLOG2	0.08	2.05	5.11
COMMENT1	0	0.22	7.65
COMMENT2	0	0.19	7.73
ABOUT	1.44	0.49	1.32
CONTACT	1.42	0.49	1.36

5.7.1.2 Frequency Domain Hierarchical Clustering Analysis Clusters Overview

Table 34 presents the clustering result from hierarchical clustering analysis. We can learn that type 1 users visit the website only once on average. They never read blog content or leave comments. They read contact and about information approximately once.

Type 2 users are the majority who read blogs on the website. They visit the website once and read each blog twice on average. In addition, type 2 users seldom leave comments or see other parts of the website.

Type 3 users are the frequent users who visit the website quite often. They visit the webpage 3.56 times on average. In addition, type 3 users read blogs 1 and blog 2 five times on average and leave seven comments on each blog.

The characteristics of user types in the hierarchical clustering result are very similar to that of the K-means clustering result.

Table 34 Frequency Domain Hierarchical Clustering Analysis Clusters Overview

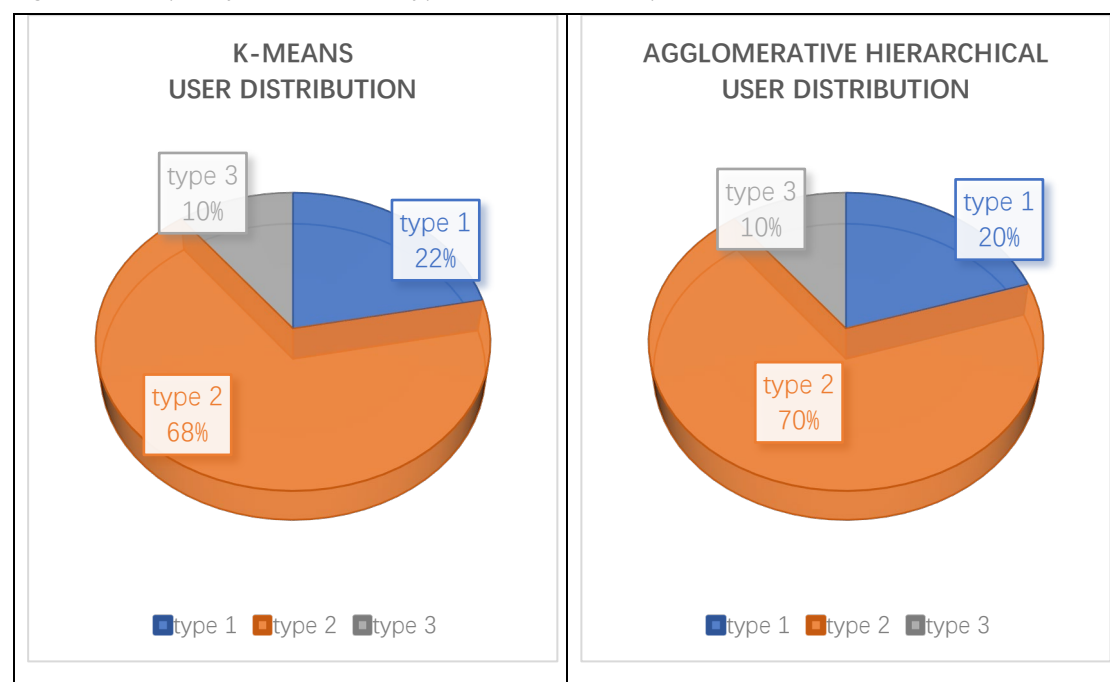
	TYPE 1	TYPE 2	TYPE 3
COUNTS	400	1392	208
FREQUENT	1	1.07	3.56
HOMEPAGE	1	1	2.56
BLOG1	0	2	5.07
BLOG2	0	2	5.11
COMMENT1	0	0.21	7.65
COMMENT2	0	0.18	7.72
ABOUT	1.48	0.5	1.32
CONTACT	1.46	0.5	1.36

5.7.2 User Type Distribution

The pie charts in Figure 29 show the partition conditions of the user types between the K-means clustering and hierarchical clustering results.

In the K-means clustering result, user type 1 has a 22% share, while in the hierarchical clustering result, user type 1 has a 20% share. 68% of the users in the K-means clustering results are user type 2, and 70% of the users in the hierarchical clustering are user type 2. The share of user type 3 is the same in both clustering results.

Figure 29 Frequency Domain User Type Distribution Comparison



5.7.3 User Type Comparison by Use Cases

We calculate the average frequency of use of the use case for each user type. The data are presented in Table 35. All values represent frequency. The capital letter 'K' in the index row represents a user type from the K-means clustering result. The capital letter 'H' in the index field indicates a user type from the hierarchical clustering result.

The same user types from the two clustering results show the same trends in the frequency of use cases.

Table 35 Frequency Domain User Type Comparison by Use Cases

	<i>Type 1</i> (K)	<i>Type 1</i> (H)	<i>Type 2</i> (K)	<i>Type 2</i> (H)	<i>Type 3</i> (K)	<i>Type 3</i> (H)
<i>Visit Home page</i>	1.0	1.0	1.0	1.0	2.6	2.6
<i>read blogs</i>	0.3	0.0	5.2	5.1	15.2	15.2
<i>leave comments</i>	0.0	0.0	0.4	0.4	15.4	15.4
<i>read contact info</i>	2.9	2.9	1.0	1.0	2.7	2.7
<i>blog 1</i>	0.1	0.0	2.0	2.0	5.1	5.1
<i>blog 2</i>	0.1	0.0	2.0	2.0	5.1	5.1
<i>comment 1</i>	0.0	0.0	0.2	0.2	7.6	7.6
<i>comment 2</i>	0.0	0.0	0.2	0.2	7.7	7.7

5.7.4 Infographic Use Case Diagrams Comparison

Infographic use case diagrams are created for each user type. The eclipses are coloured and indicate how often users of a particular user type act within the context of that use case. Similar to the infographic use case diagrams created in the duration domain cluster analyses, the redder the colours within the eclipses, the more often users use the functionalities under the use cases shown. On the contrary, the greener the colours within the eclipses, the less frequent users use the functionalities under the depicted use cases. The infographic use case diagrams of three user types from the results of K-means clustering and hierarchical clustering are shown in Table 36.

Based on the colour within the eclipses, you can see that users of type 1 of the K-means clustering and the hierarchical clustering hardly use any functions of the blog system except for the 'read contact information'.

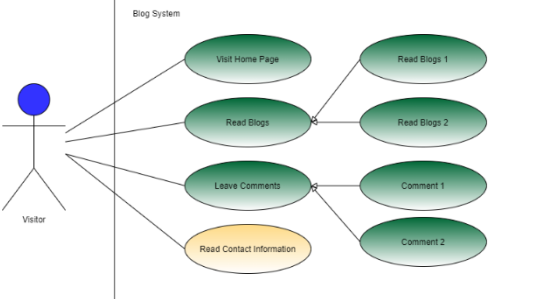
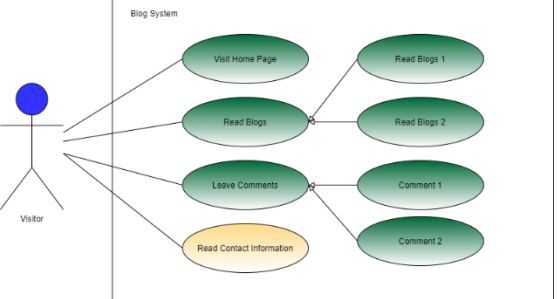
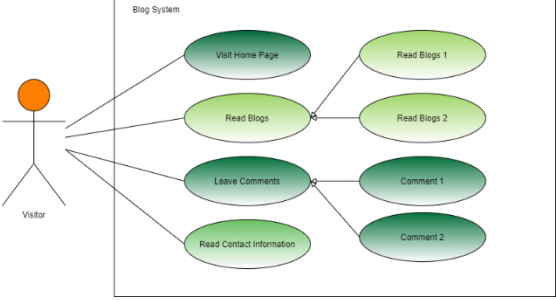
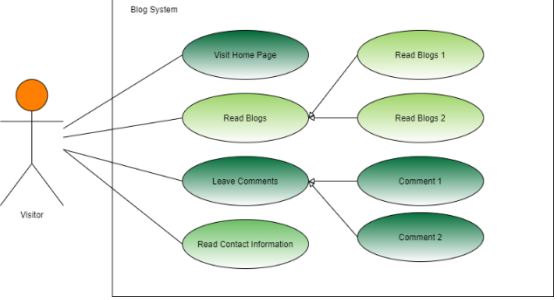
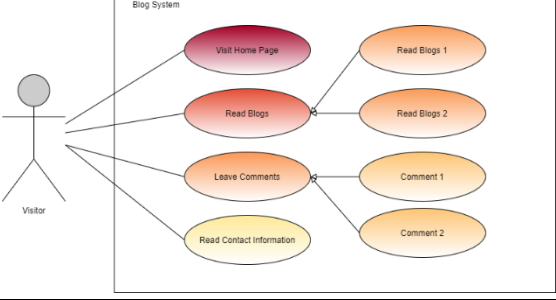
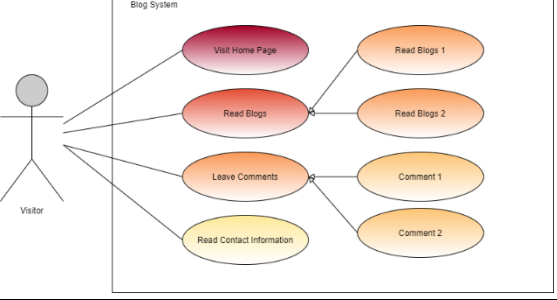
Type 2 users of both clustering results use 'read blogs' slightly more than Type 1. As can be seen in the use case diagram for Type 2, the colours in the eclipses representing 'read blogs'

and 'blog 1', 'blog 2' are lighter than those of Type 1.

Users of type 3 of both clustering results often use all features, as all eclipses have red hues.

We can assume that type 3 users are professional users who visit the system frequently. They also read blogs and leave comments regularly.

Table 36 Frequency Domain Infographic Use Case Diagrams Comparison

	K-means clustering	Agglomerative Hierarchical clustering
User Type 1		
User Type 2		
User Type 3		

5.7.5 *Conclusions in Frequency Domain*

5.7.5.1 **System Improvement Suggestions**

In general, the majority of visitors access blog content in low frequency without leaving comments. A small number of visitors access all features very frequently. We suggest that the blog system navigates visitors to the blog content on the home page. This way, most blog users have quick access to the blog content that interests them.

Type 3 users are professional users who regularly read blogs and make comments. Therefore, we can develop features that notify professional users about the latest updates to blog content and comments. We can also forward blog content to these users so that they do not have to keep visiting the home page for blog updates.

5.7.5.2 **Clustering Algorithm Comparison**

In the frequency domain, the K-means algorithm and the agglomerative hierarchical algorithm produce very identical results. Their user types have similar behaviour patterns. From the perspective of profile results, these two algorithms perform equally well in generating user profiles in the frequency domain.

5.8 Subtype Discovery

The dendrogram created in hierarchical clustering makes it possible to identify an appropriate number of clusters in cluster analysis. The dendrogram also shows a hierarchical structure of the cluster aggregation process. In hierarchical clustering in the frequency domain, we draw a horizontal line across three vertical lines. If we move the horizontal line down, we can see that two clusters split into four subsequent clusters. Accordingly, these four subsequent clusters are the user subtypes that make up the larger user type we have in hierarchical clustering in the frequency domain.

The new horizontal line has five intersections in the dendrogram. To discover details in these subsequent clusters, we run the agglomerative hierarchical clustering instance again, setting the number of clusters to five. We compare the new clustering result with the original one in detail.

.

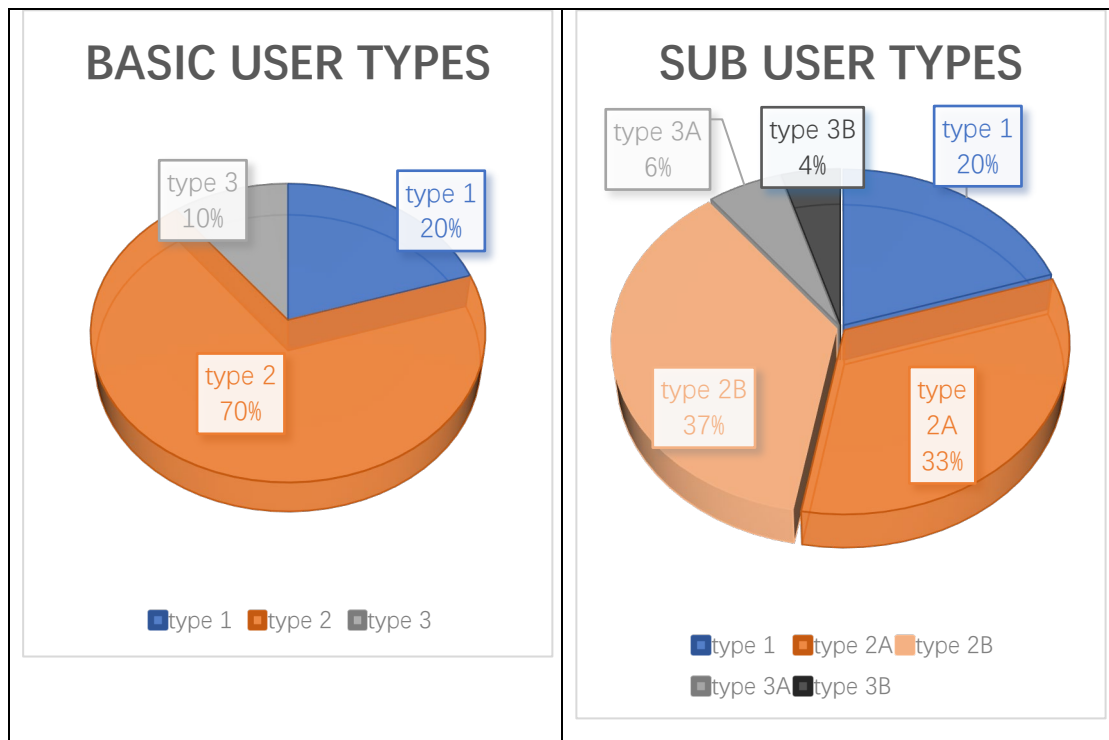
5.8.1 User Type Distribution

In identifying the subtypes, user type 2 and user type 3 are divided into four additional user types. Depending on their parent user types, we name these four subsequent user types as 'type 2A', 'type 2B', 'type 3A', 'type 3B'. User type 2A and 2B are the subtypes of user type 2. User type 3A and 3B are the subtypes of user type 3. Figure 30 showcases the user type distribution condition.

User type 2A accounts for 33% of users, while user type 2B accounts for 37%. Their sum is equal to the share that their parent user type 'Type 2' has.

User type 3A and type 3B are the subsequent user types of type 3. 6% of the users are type 3A, and 4% of the users are type 3B. These two subsequent user types make up user type 3.

Figure 30 Hierarchical Clustering for Subtypes User Type Distribution Comparison



5.8.2 User Type Comparison by Use Cases

As with the calculation for the comparison of frequency domain clustering results, we calculate the average frequency of use of the use case by the following user types. The data are presented in Table 37. All values represent frequency. The columns in gray contain the data for the subsequent user types 2A, 2B, 3A, and 3B.

In general, type 2 and subsequent types 2A and 2B are the users who read blogs regularly. From table 37, we can learn that the difference between type 2A and type 2B is their average frequency of leaving comments. Type 2B users leave comments more frequently than Type 2A users. The average frequency scores of these two subtypes range from 0.7 to 0.

Type 3 and its subsequent types 3A and 3B are the professional users who read blogs a lot and leave many comments. Type 3A users visit the blog system more frequently than type 3B. On average, the frequency value of type 3A is 50% higher than that of type 3B. Therefore, we can conclude that type 3A users are the most 'professional' users among all user types.

Table 37 Subtype Discovering User Type Comparison by Use Cases

	<i>Type 1</i>	<i>Type 2</i>	<i>Type 2a</i>	<i>Type 2b</i>	<i>Type 3</i>	<i>Type 3a</i>	<i>Type 3b</i>
<i>Visit Home page</i>	1.0	1.0	1.0	1.0	2.6	<u>3.0</u>	<u>2.0</u>
<i>read blogs</i>	0.0	5.1	5.0	5.2	15.2	<u>17.7</u>	<u>12.0</u>
<i>leave comments</i>	0.0	0.4	<u>0.0</u>	<u>0.7</u>	15.4	<u>17.9</u>	<u>12.1</u>
<i>read contact info</i>	2.9	1.0	<u>0.5</u>	<u>1.5</u>	2.7	<u>3.1</u>	<u>2.1</u>
<i>blog 1</i>	0.0	2.0	2.0	2.0	5.1	<u>5.9</u>	<u>4.0</u>
<i>blog 2</i>	0.0	2.0	2.0	2.0	5.1	<u>6.0</u>	<u>4.0</u>
<i>comment 1</i>	0.0	0.2	0.0	0.4	7.6	<u>8.8</u>	<u>6.2</u>
<i>comment 2</i>	0.0	0.2	0.0	0.3	7.7	<u>9.1</u>	<u>6.0</u>

5.8.3 Infographic Use Case Diagram Showcase

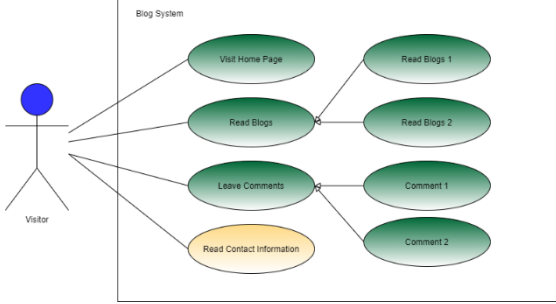
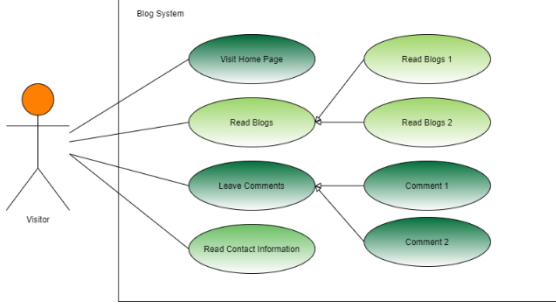
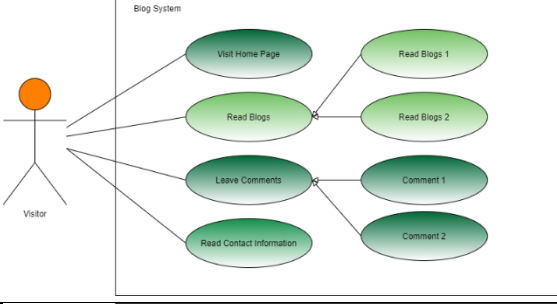
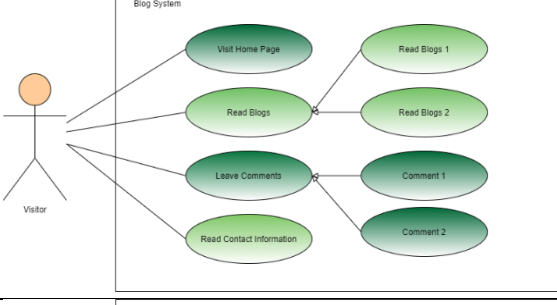
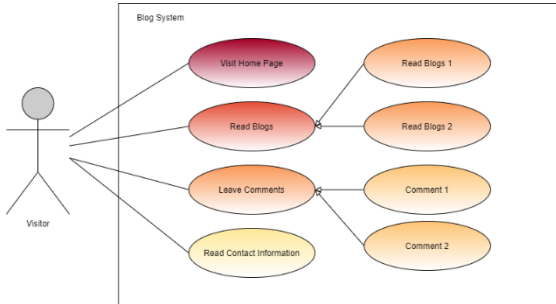
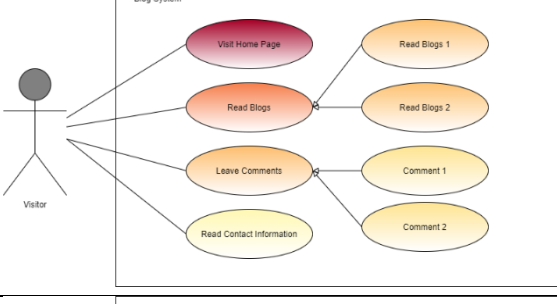
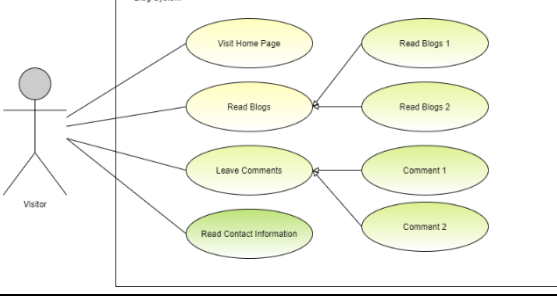
We created infographic use case diagrams for the following four user types we discovered. We compare the diagrams of the subtypes to the diagrams of the basic user types. We merge these diagrams into the same table. In this way, people can visually understand the inferred relationship of these user types using Table 38.

On the left side of the table, we list the use case diagrams of the main user types. The use case diagrams of the subsequent user types are listed next to their parent types on the right side of the table. We can see that subtypes and their parent types have a similar colour pattern, which means that the users of these types have similar behaviour patterns. By increasing the cluster numbers in the hierarchical cluster analysis, we can differentiate the subsequent user types that are derived from the main user types.

For example, the colours in the use case diagrams of Type 3A and Type 3B are shades of red. However, the colours of the eclipses in the use case diagrams of Type 3B are brighter. Such a

result indicates that Type 3B users are less active than Type 3A users.

Table 38 Frequency Domain Subtype Discovering Infographic Use Case Diagrams Comparison

	Main Types	Subtypes
User Type 1	 <p>The diagram shows a stick figure actor labeled 'Visitor' connected to four use cases: 'Visit Home Page', 'Read Blogs', 'Leave Comments', and 'Read Contact Information'. 'Visit Home Page' is connected to 'Read Blogs 1' and 'Read Blogs 2'. 'Read Blogs' is connected to 'Read Blogs 1' and 'Read Blogs 2'. 'Leave Comments' is connected to 'Comment 1' and 'Comment 2'. 'Read Contact Information' is connected to 'Comment 1' and 'Comment 2'. The use cases are colored in shades of green and yellow.</p>	
User Type 2	 <p>The diagram shows a stick figure actor labeled 'Visitor' connected to four use cases: 'Visit Home Page', 'Read Blogs', 'Leave Comments', and 'Read Contact Information'. 'Visit Home Page' is connected to 'Read Blogs 1' and 'Read Blogs 2'. 'Read Blogs' is connected to 'Read Blogs 1' and 'Read Blogs 2'. 'Leave Comments' is connected to 'Comment 1' and 'Comment 2'. 'Read Contact Information' is connected to 'Comment 1' and 'Comment 2'. The use cases are colored in shades of green and yellow.</p>	 <p>The diagram shows a stick figure actor labeled 'Visitor' connected to four use cases: 'Visit Home Page', 'Read Blogs', 'Leave Comments', and 'Read Contact Information'. 'Visit Home Page' is connected to 'Read Blogs 1' and 'Read Blogs 2'. 'Read Blogs' is connected to 'Read Blogs 1' and 'Read Blogs 2'. 'Leave Comments' is connected to 'Comment 1' and 'Comment 2'. 'Read Contact Information' is connected to 'Comment 1' and 'Comment 2'. The use cases are colored in shades of green and yellow.</p>  <p>The diagram shows a stick figure actor labeled 'Visitor' connected to four use cases: 'Visit Home Page', 'Read Blogs', 'Leave Comments', and 'Read Contact Information'. 'Visit Home Page' is connected to 'Read Blogs 1' and 'Read Blogs 2'. 'Read Blogs' is connected to 'Read Blogs 1' and 'Read Blogs 2'. 'Leave Comments' is connected to 'Comment 1' and 'Comment 2'. 'Read Contact Information' is connected to 'Comment 1' and 'Comment 2'. The use cases are colored in shades of green and yellow.</p>
User Type 3	 <p>The diagram shows a stick figure actor labeled 'Visitor' connected to four use cases: 'Visit Home Page', 'Read Blogs', 'Leave Comments', and 'Read Contact Information'. 'Visit Home Page' is connected to 'Read Blogs 1' and 'Read Blogs 2'. 'Read Blogs' is connected to 'Read Blogs 1' and 'Read Blogs 2'. 'Leave Comments' is connected to 'Comment 1' and 'Comment 2'. 'Read Contact Information' is connected to 'Comment 1' and 'Comment 2'. The use cases are colored in shades of red and orange.</p>	 <p>The diagram shows a stick figure actor labeled 'Visitor' connected to four use cases: 'Visit Home Page', 'Read Blogs', 'Leave Comments', and 'Read Contact Information'. 'Visit Home Page' is connected to 'Read Blogs 1' and 'Read Blogs 2'. 'Read Blogs' is connected to 'Read Blogs 1' and 'Read Blogs 2'. 'Leave Comments' is connected to 'Comment 1' and 'Comment 2'. 'Read Contact Information' is connected to 'Comment 1' and 'Comment 2'. The use cases are colored in shades of red and orange.</p>  <p>The diagram shows a stick figure actor labeled 'Visitor' connected to four use cases: 'Visit Home Page', 'Read Blogs', 'Leave Comments', and 'Read Contact Information'. 'Visit Home Page' is connected to 'Read Blogs 1' and 'Read Blogs 2'. 'Read Blogs' is connected to 'Read Blogs 1' and 'Read Blogs 2'. 'Leave Comments' is connected to 'Comment 1' and 'Comment 2'. 'Read Contact Information' is connected to 'Comment 1' and 'Comment 2'. The use cases are colored in shades of green and yellow.</p>

5.8.4 Subtype Discovering Process Summary

Finding subsequent user types helps people learn the composition of basic user types with more details. The blog system can provide more customized content to each user type. Hierarchical clustering is a perfect tool for discovering user types. We could keep increasing the number of clusters following the dendrogram until we find enough information about the desired user behaviour.

5.9 Clustering Algorithm Choice Discussion

From the comparison of the results of the cluster analyses in the duration and frequency domains, it is clear that the K-means clustering algorithm and the agglomerative hierarchical clustering algorithm give very similar clustering results when we set the same value for the number of clusters. In other words, both algorithms are very similar in terms of the quality of the results. However, the K-means clustering provides better computational efficiency.

In terms of computational efficiency, the K-means clustering method consumes less time than the agglomerative hierarchical clustering method in the same user profiling task. The differences in time consumption are shown in Table 39. The data are recorded by the software 'Jupyter notebook,' which is used to perform all cluster analyses in this work. The values in the table indicate the duration of the cluster analysis processes. We know that agglomerative hierarchical clustering takes more than two times longer than K-means clustering. Therefore, we recommend using K-means clustering when creating user profiles to maximize computational efficiency.

Table 39 Time Consumption of Executing Two Clustering Algorithms

	K-means Clustering	Agglomerative Hierarchical Clustering
Duration Domain	59ms	144ms
Frequency Domain	17ms	62ms

The agglomerative hierarchical clustering algorithm is the best option for user profiling based on use cases. There are several reasons for this. First, although K-means clustering is faster, the time required for agglomerative hierarchical clustering is of the same order as K-means clustering. Such a difference does not really matter for user profiling.

Second, hierarchical clustering provides a solution to subdivide basic user types into

subsequent user types. Such a feature allows analysts to adjust the scope of the information collected. From a product management perspective, it can be used to pinpoint a user type that uses a particular functionality to a certain degree. Let us take the hierarchical cluster analysis of the Discover Frequency Domain subtype as an example. We are supposed to find a group of users who only read blogs without doing anything else. We cannot discover them by learning the basic user types. We can increase the number of clusters following the dendrogram until another user type emerges that meets our requirements. Then we can analyse such a user type in detail. The last reason to choose hierarchical clustering is its strong visualization capability. A dendrogram is a perfect tool to demonstrate the inferred relationship between user types. One can learn how basic user types break down into subsequent user types.

5.10 Chapter Summary

In this chapter, we compare the results of user profiling. Three sets of comparisons proved that it is possible to group users into instinct user types using our use case based user profiling method. The infographic uses case diagrams to optimize colours to visualize the differences between user types. It performs the function of visualizing the structure of system functionality from basic use case diagrams. It also shapes unique user profiles based on use cases.

We also prove that hierarchical clustering is better for building user profiles based on use cases. The hierarchical clustering process gives users access to subsequent user type discovery. This allows learning how small groups of users make up the overall user base and how users of subtypes interact with the system. Such a feature can only be achieved by applying hierarchical clustering methods.

We hope that our methodology will soon be applied to the evaluation of software products in practice. We anticipate that a more automated user profiling system will be developed in the future. In the next chapter, we discuss the design of an interactive user profiling system. Such a system can help users monitor system usage conditions from a use case perspective.

6 Integrated Tool Support for Use Case Based User Profiling

The application of the use case based user profiling methodology described in this thesis was enabled by a combination of off-the-shelf available programs and custom-built implementations. However, some of the steps in applying our methodology required manual intervention, such as data pre-processing, cluster analysis, and creating the infographic use case diagrams. This chapter discusses a design for an integrated interactive tool that would provide a turnkey solution for applying our methodology in real-world situations.

Human analysts must manage the data collection and cluster analysis through programming. Most importantly, humans need to manually investigate the dendrogram and set the number of clusters to discover the subsequent user types. An automated user profiling tool should complete the following tasks. First, it should create use case diagrams for the software or import existing use case diagrams from UML design tools. Next, it imports usage data that can be mapped to use cases and features from software log files or a database. Afterwards, the user profiling tool should start a use case based cluster analysis program which may group users into clusters. Finally, the tool can create infographic use case diagrams automatically. People can visit the tool to check how different types of users are using the current software, the subtypes of users, and know how to improve the software in the next release. The following sections introduce the most important features for the future use case based user profiling tools.

6.1 Interactive Features

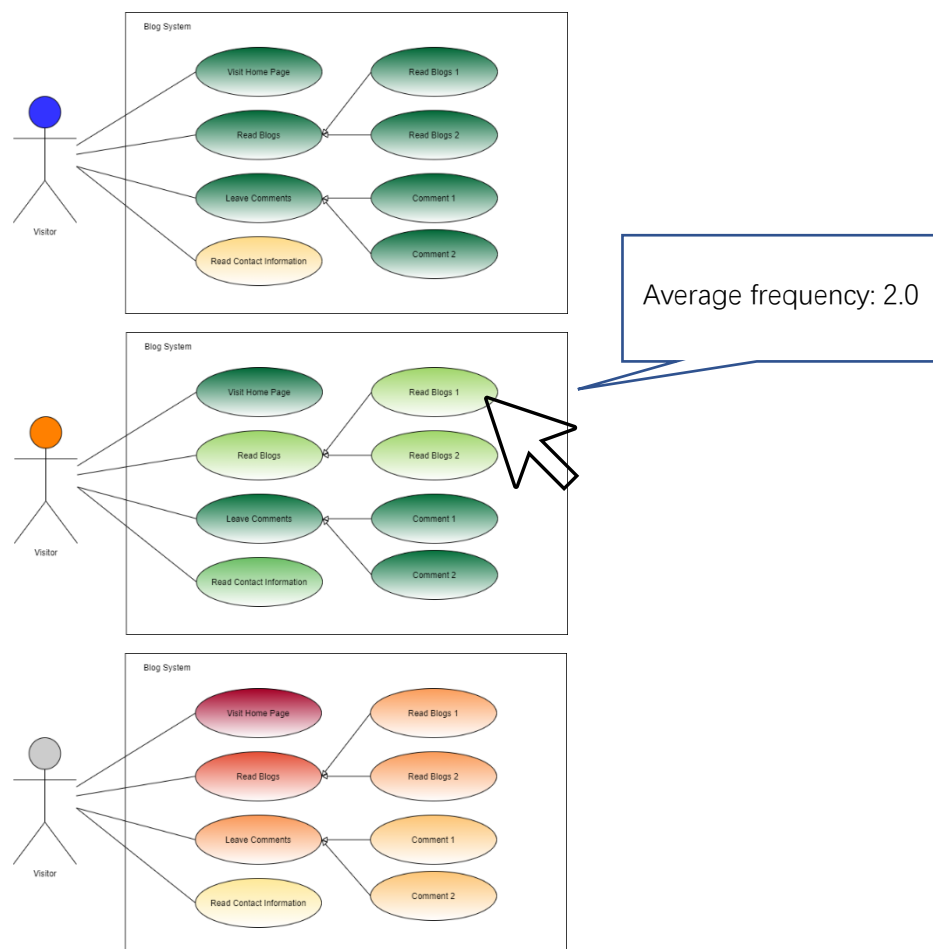
The interactive user profiling system automatically performs user cluster analysis and presents the results visually. The system provides analysts with an interactive interface to view, interpret, and comment on the infographic use case diagrams generated from the cluster analyses.

The essential function of the system is automated user profiling. Analysts can select a time range on the system interface for user profiling. The system then retrieves software usage data in that time period from the target software product and converts that data into manageable data sets. The dataset should be prepared according to the requirements of use case-based user profiling. The usage data collected must be mapped to software use cases.

Agglomerative hierarchical clustering is then performed in the background, and the cluster analysis results, including infographic use case diagrams, are produced. People who use this user profiling tool can choose from the options the system suggests. They can decide to investigate primary user types or discover subsequent user types. People can interact with diagrams to learn the information they need. When they request the user profiling results, they get the basic user type information. If they want to learn more about the subtypes, click on the use case diagrams for the primary user type. Then the use case diagrams for the user subtypes will show up.

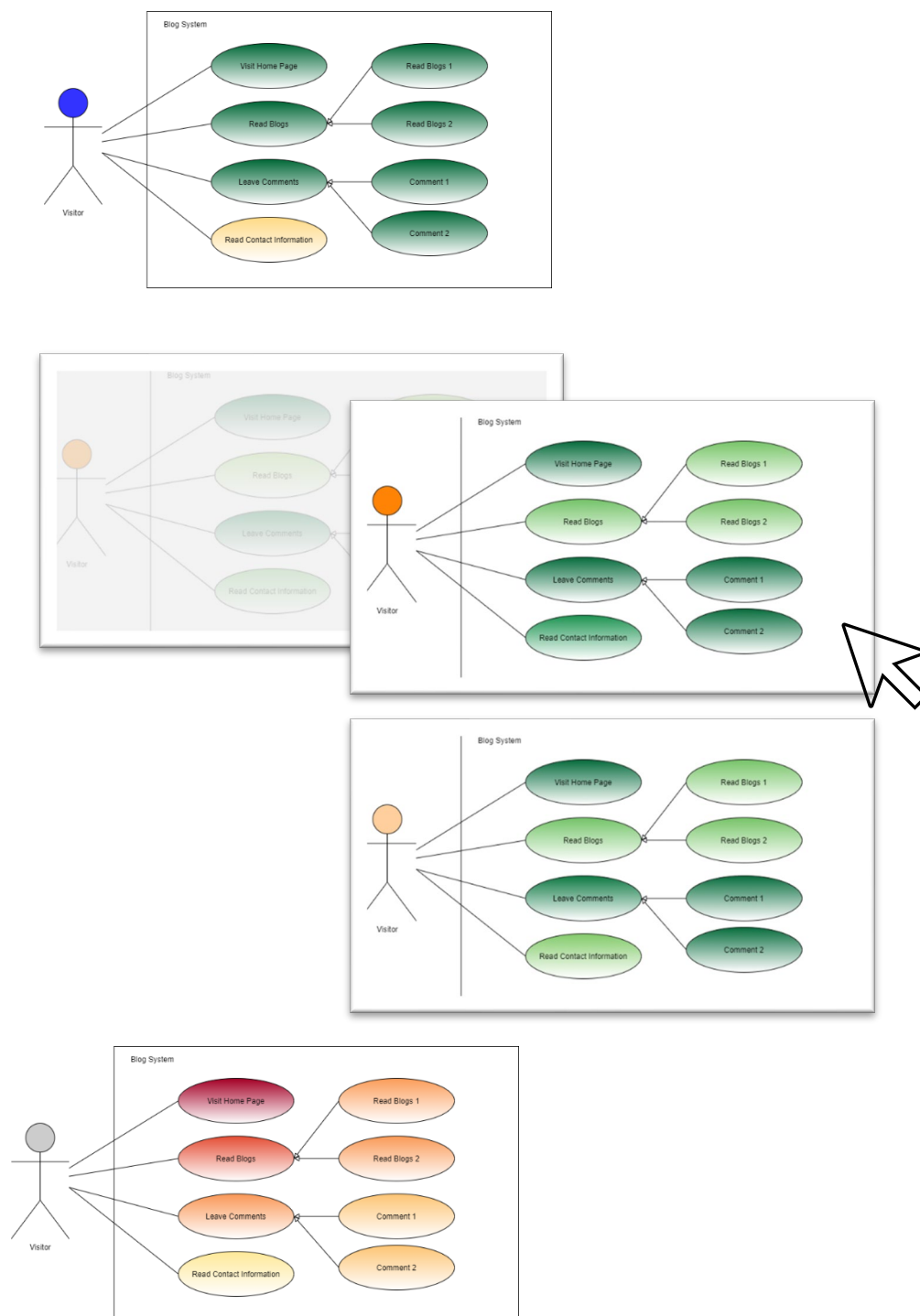
Let us take the hierarchical cluster analysis in the Duration results as an example. We design a user interface as shown in Figure 31, and when users want to see the user profiling results, the use case diagrams for the basic user types appear. Users can hover the mouse over the eclipse that represents the use cases. A text box indicates how often users of that user type visit that use case on average.

Figure 31 Interactive User Profiling System User Interface Design



If anyone is interested in the following user types, click on the use case diagram. Such an action is the action that activates the subtype detection mode. The system increases the number of clusters and repeats the process of user profiling. After a short time, the use case diagrams of the subsequent user types are displayed on the user interface. As shown in Figure 32, users can see two unique infographic use case diagrams representing two successive user types.

Figure 32 Interactive User Profiling System User Interface Design with Subsequent User Types



6.2 The Feature of Improving Software Use Case Models Automatically

The future use case based user profiling system should have the feature of enriching the original use case models of the system. It will help people update the software use case model

automatically. The current user profiling system generates a series of infographic use case diagrams given a forward engineered software use case model. These use case models we defined before the user profiling process are human-made. The future user profiling system shall enrich the use case models with cluster analyses results. Use cases not frequently accessed by users should be highlighted and considered removed, while use cases having subsets should be extended. The new extended use cases can be highlighted in the user profiling tool, and people can decide to push these changes back into a UML tool. The updated use case diagram may become the basis for updating product management software. The use case-based user profiling system would continuously monitor software usage data in the future. For example, we suppose there would be a change in user behaviour patterns for unknown reasons. Then, in a scheduled use case based user profiling analysis, the system detects a change in use cases' hierarchical structure. The system would now compare the latest use case structure to the last version, take action accordingly, and replace the previous use case subsets with the newly discovered ones. Although the system may not name these newly discovered use cases, it can notify the system administrators of the changes. The future system may also generate reports regularly that review the software feature usage condition. When new software features go online, use cases representing these features will be added to the software use case model. The future user profiling system may output reports indicating how end-users invoke these new use cases and features. For example, the report may tell people how each user segment uses these new features and how user segments are formed. The last point is that the future user profiling system can detect redundant use cases and remove them from the use case model. For example, the system proceeds a use case based user profiling analysis given an obsolete software use case model. Then, the user profiling system may detect some use cases never activated by any user type. Afterwards, the profiling system can remove these use cases from the use case model since these interactions no longer occur.

6.3 Summary

In this chapter, we present a sketch of the design of the interactive user profile system. In this design, users can view statistical data about the user type on infographic use case diagrams.

This future tool allows people to learn about a user type quantitatively and qualitatively simultaneously. It also simplifies the search for additional user types. They do not need data mining skills or programming knowledge to perform cluster analysis. With clicks, they can learn how users are grouped into instinct types and how basic user types break down into subsequent user types. Due to the time limit, this design remains on paper.

Additionally, the future use case based user profiling tools can improve the software use case model itself. It would automatically optimize the software use case model by analysing system usage data. We hope that such an interactive user profiling system will soon enter the real world and bring convenience to people.

7 Conclusions

We have defined a use case based user profiling methodology to support software product management. It describes a framework for user profiling based on data mining of actual software usage patterns of deployed software. Based on use case models that were created during the design of a software system, it identifies groups of users with similar behaviour patterns. In addition, it enables the discovery of new user groups and extended use case models. Our research effectively used open-source datasets to study the user behaviour of specific software products and developed a method to simulate user inputs for the software products under evaluation. We evaluated several clustering methods for efficiency and functionality in the context of our methodology. The K-means clustering algorithm has its advantage in computational efficiency. Hierarchical clustering is the best option to identify user types and the extended use cases that shape these primary user groups. Using the information provided by the dendrogram, one can understand the inferred relationship between individual user types and discover new user segments.

In this research project, we developed a methodology that extends the concept of software usage data mining: the Use Case Based User Profiling methodology. This profiling process aims to use the use case model as a top design for data understanding and data preparation in data mining related projects. By applying this methodology in software product management, we can group software end-users into instinct user types and learn how users of each user type use software functionalities. We also prove that mapping software usage data to use cases is possible. We can create user profiles using the use case interaction duration or frequency data.

The design of infographic use case diagrams is an extension of existing use case diagrams that have been shown to be an effective visualization for user profiling. These diagrams show the conditions of software usage by different types of users, using colour and annotations to visually distinguish user types. Readers can see the differences between user types by the shades of the colours in the eclipses. We use a reddish-green colour scheme to represent the usage state. A redder colour represents more prolonged usage or a higher frequency of usage,

and a greener colour means the opposite.

The experiment of applying the use case based user profiling to a web blog application demonstrated that our extended use case models based on data mining of actual software usage could improve the quality of software product management. Use case models serve as design tools in the early stages of software development and as a medium to help product managers understand how their customers use software products. This research has contributed to the further application of use case models.

Our research used a combination of available off-the-shelf tools and custom-built implementations. Several steps of our methodology required manual intervention. Based on our experiences and results, we have described a design for an integrated tool that would support our methodology in an interactive fashion. Such a tool would enable incremental improvement of software releases based on explorative evaluation of actual software usage patterns and involve the tool user only in the creative human decisions required in planning future releases. In addition, it would report on the effectiveness of new releases in terms of user usage patterns.

Future research is needed to explore the impact of combining duration and frequency data in user profiling and consider which other factors are effective indicators that help in evaluating functional software usage. Furthermore, our profiling method could be extended to cover additional UML design models, such as Class and Activity models.

In addition, we would like to investigate how K-means clustering and hierarchical clustering algorithms perform with a dataset that contains values of different dimensions.

Finally, we would like to test our methodology in an enterprise environment on deployed applications with an actual user base rather than a simulated one. To achieve such a goal, steps towards implementing a more integrated tool to support our methodology would be needed. This would also require investigating and implementing non-invasive instrumentation mechanisms for recording software usage at the functional level. This would then enable us to evaluate the real-world effectiveness of our use case based user profiling methodology for software developers and product managers in software evolution.

8 Reference

1. Morales-Ramirez, I., Perini, A., & Guizzardi, R. S. (2015). An ontology of online user feedback in software engineering. *Applied Ontology*, 10(3-4), 297-330.
2. Guzman, E., Bhuvanagiri, P., & Bruegge, B. (2014, September). Fave: Visualizing user feedback for software evolution. In *2014 Second IEEE Working Conference on Software Visualization* (pp. 167-171). IEEE.
3. Alexander, J., Cockburn, A., & Lobb, R. (2008). AppMonitor: A tool for recording user actions in unmodified Windows applications. *Behaviour Research Methods*, 40(2), 413-421.
4. Jiang, Y., & Yu, S. (2008, January). Mining e-commerce data to analyse the target customer behaviour. In *First International Workshop on Knowledge Discovery and Data Mining (WKDD 2008)* (pp. 406-409). IEEE.
5. Su, Y. S., & Wu, S. Y. (2021). Applying data mining techniques to explore user behaviours and watching video patterns in converged IT environments. *Journal of Ambient Intelligence and Humanized Computing*, 1-8.
6. Syadzali, C., Suryono, S., & Suseno, J. E. (2020). Business Intelligence using the K-Nearest Neighbor Algorithm to Analyse Customer Behaviour in Online Crowdfunding Systems. In *E3S Web of Conferences* (Vol. 202, p. 16005). EDP Sciences.
7. Godbole, S., & Roy, S. (2008, August). Text classification, business intelligence, and interactivity: automating c-sat analysis for services industry. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 911-919).
8. Cheng, Z., Caverlee, J., & Lee, K. (2010, October). You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (pp. 759-768).
9. Pennacchiotti, M., & Popescu, A. M. (2011, July). A machine learning approach to twitter user classification. In *Fifth international AAAI conference on weblogs and social media*.
10. Guimaraes, R. G., Rosa, R. L., De Gaetano, D., Rodriguez, D. Z., & Bressan, G. (2017). Age groups classification in social network using deep learning. *IEEE Access*, 5, 10805-10816.
11. Qazi, A. M., Rauf, A., & Minhas, N. M. (2016). A Systematic Review of Use Cases based Software Testing Techniques. *International Journal of Software Engineering and Its Applications*, 10(11), 337-360.
12. Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum* (Vol. 1). Upper Saddle River: Prentice Hall.
13. Krusche, S., Alperowitz, L., Bruegge, B., & Wagner, M. O. (2014, June). Rugby: an agile process model based on continuous delivery. In *Proceedings of the 1st International*

Workshop on Rapid Continuous Software Engineering (pp. 42-50).

14. Pachidi, S., Spruit, M., & Van De Weerd, I. (2014). Understanding users' behaviour with software operation data mining. *Computers in Human Behaviour*, 30, 583-594.
15. Nasraoui, O., Soliman, M., Saka, E., Badia, A., & Germain, R. (2007). A web usage mining framework for mining evolving user profiles in dynamic web sites. *IEEE transactions on knowledge and data engineering*, 20(2), 202-215.
16. Cooley, R., Mobasher, B., & Srivastava, J. (1997, November). Web mining: Information and pattern discovery on the world wide web. In *Proceedings ninth IEEE international conference on tools with artificial intelligence* (pp. 558-567). IEEE.
17. Wang, G., Konolige, T., Wilson, C., Wang, X., Zheng, H., & Zhao, B. Y. (2013). You are how you click: Clickstream analysis for sybil detection. In *22nd {USENIX} Security Symposium ({USENIX} Security 13)* (pp. 241-256).
18. Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 75-105.
19. Booch, G. (2005). *The unified modeling language user guide*. Pearson Education India.
20. Dobing, B., & Parsons, J. (2006). How UML is used. *Communications of the ACM*, 49(5), 109-113.
21. Specification, O. A. (2007). *Omg unified modeling language (omg uml), superstructure, v2. 1.2*. Object Management Group, 70.
22. El-Ramly, M., & Stroulia, E. (2004, May). Mining Software Usage Data. In *MSR* (pp. 64-68).
23. Junco, R. (2013). Comparing actual and self-reported measures of Facebook use. *Computers in Human Behaviour*, 29(3), 626-631.
24. Germanakos, P., Tsianos, N., Lekkas, Z., Mourlas, C., & Samaras, G. (2008). Capturing essential intrinsic user behaviour values for the design of comprehensive web-based personalized environments. *Computers in Human Behaviour*, 24(4), 1434-1451.
25. El-Ramly, M., Stroulia, E., & Sorenson, P. (2002, July). Recovering software requirements from system-user interaction traces. In *Proceedings of the 14th international conference on Software engineering and knowledge engineering* (pp. 447-454).
26. Kanoje, S., Girase, S., & Mukhopadhyay, D. (2015). User profiling trends, techniques and applications. *arXiv preprint arXiv:1503.07474*.
27. Poo, D., Chng, B., & Goh, J. M. (2003, January). A hybrid approach for user profiling. In *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the* (pp. 9-pp). IEEE.
28. Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis* (Vol. 344). John Wiley & Sons.

29. Trebuňa, P., Halčinová, J., Fil'o, M., & Markovič, J. (2014, January). The importance of normalization and standardization in the process of clustering. In 2014 IEEE 12th International Symposium on Applied Machine Intelligence and Informatics (SAMI) (pp. 381-385). IEEE.
30. Gentleman, R., & Carey, V. J. (2008). Unsupervised machine learning. In Bioconductor case studies (pp. 137-157). Springer, New York, NY.
31. MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297).
32. Lloyd, S. (1982). Least squares quantization in PCM. IEEE transactions on information theory, 28(2), 129-137.
33. Mohamad, I. B., & Usman, D. (2013). Standardization and its effects on K-means clustering algorithm. Research Journal of Applied Sciences, Engineering and Technology, 6(17), 3299-3303.
34. Tanioka, K., & Yadohisa, H. (2012). Effect of data standardization on the result of k-means clustering. In Challenges at the Interface of Data Analysis, Computer Science, and Optimization (pp. 59-67). Springer, Berlin, Heidelberg.
35. Bhandari, A. (2021). Feature scaling: Standardization vs normalization. Analytics Vidhya. Retrieved November 11, 2021, from <https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/>

9 List of Tables

Table 1 Service Desk Dataset Variables Overview	25
Table 2 Service Desk IT System Dataset Overview	27
Table 3 Service Desk K-means Clustering Centroids Details	28
Table 4 Cluster Details of the Service Desk Cluster Analysis.....	29
Table 5 K-means Clustering Result Overview with Normalized Data	31
Table 6 Incidents Characteristics in the Clusters.....	32
Table 7 Duration Domain Dataset Values Explanation.....	45
Table 8 Frequency Domain Dataset Values Explanation	46
Table 9 Blog System Use Case Explanation.....	48
Table 10 Use Case – Values Mapping Table	49
Table 11 Use Case Subsets-Value Mapping Table	49
Table 12 Duration Domain K-means Clustering Cluster Overview	51
Table 13 Duration Domain Use Case Based User Profiles.....	51
Table 14 Duration Domain K-means Clustering Clusters Use Case Usage Degree Normalized.....	52
Table 15 Duration Domain Infographic Use Case Diagram of K-means Clustering.....	53
Table 16 Duration Domain Hierarchical Clustering Clusters Overview.....	56
Table 17. Duration Domain Hierarchical Clustering Clusters Use Case Usage Degree Normalized.....	56
Table 18 Duration Domain Infographic Use Case Diagram of Hierarchical Clustering...	58
Table 19 Frequency Domain K-means Clustering Cluster Overview.....	60
Table 20 Frequency Domain Use Case Based User Profiles	60
Table 21. Frequency Domain K-means Clustering Clusters Use Case Usage Degree Normalized.....	61
Table 22 Frequency Domain Infographic Use Case Diagram of K-means Clustering.....	62

Table 23 Frequency Domain Hierarchical Clustering Clusters Overview.....	65
Table 24 Frequency Domain Hierarchical Clustering Clusters Use Case Usage Degree Normalized.....	65
Table 25 Frequency Domain Infographic Use Case Diagram of Hierarchical Clustering	67
Table 26 Frequency Domain Hierarchical Clustering for Subtypes Clusters Overview ...	69
Table 27 Frequency Domain Hierarchical Clustering for Subtypes Clusters Use Case Usage Degree Normalized	70
Table 28 Frequency Domain Infographic Use Case Diagram of Subtype Discovery	71
Table 29 Duration Domain K-means Clustering Analysis Clusters Overview	73
Table 30 Duration Domain Hierarchical Clustering Analysis Clusters Overview.....	74
Table 31 Duration Domain User Type Comparison By Use Cases	75
Table 32 Duration Domain Infographic Use Case Diagram Comparison	77
Table 33 Frequency Domain K-means Clustering Analysis Clusters Overview.....	79
Table 34 Frequency Domain Hierarchical Clustering Analysis Clusters Overview.....	80
Table 35 Frequency Domain User Type Comparison by Use Cases	81
Table 36 Frequency Domain Infographic Use Case Diagrams Comparison	82
Table 37 Subtype Discovering User Type Comparison by Use Cases.....	85
Table 38 Frequency Domain Subtype Discovering Infographic Use Case Diagrams Comparison.....	86
Table 39 Time Consumption of Executing Two Clustering Algorithms	87

10 List of Figures

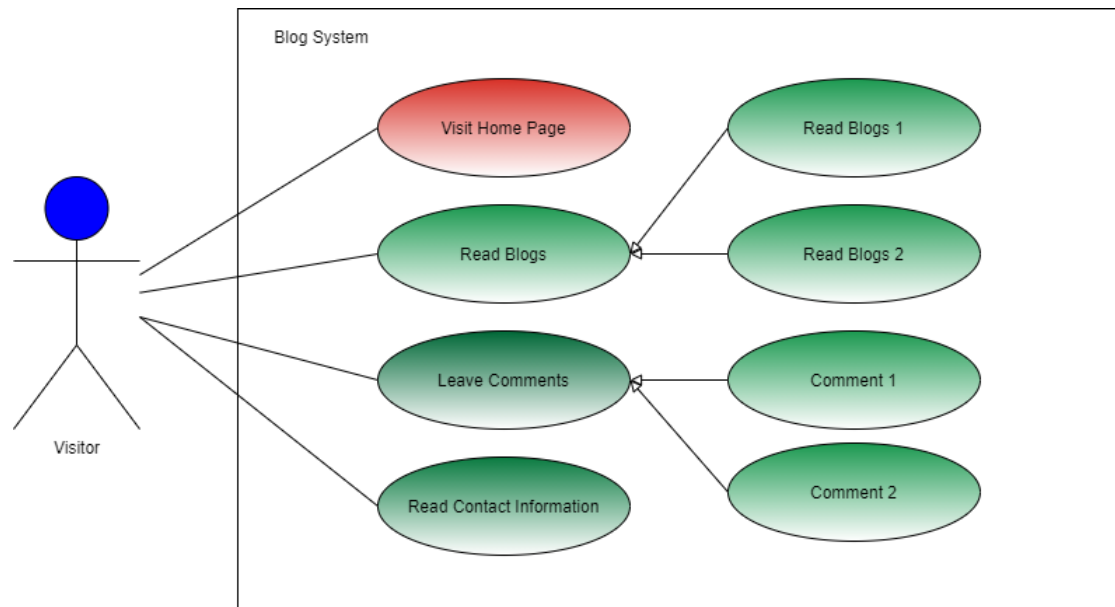
Figure 1 An Example of a Use Case Diagram	13
Figure 2 Phases of the Current CRISP-DM Process Model for Data Mining	15
Figure 3 Use Case Based User Profiling Process Model.....	21
Figure 4 Service Desk IT System Use Case Diagram	26
Figure 5 SSE as a Function of the Number of Clusters.....	28
Figure 6 Service Desk Infographic Use Case Diagram.....	30
Figure 7 Service Desk Infographic Use Case Diagram Part 2.....	33
Figure 8 Ratio Based Infographic Use Case Diagram Design	34
Figure 9 Experiment Software Components Structure	36
Figure 10 Blog System Use Case Diagram	39
Figure 11 Blog System Homepage.....	40
Figure 12 Blog System Blog Content Entrances	40
Figure 13 Blog System Blog Content Main Page	41
Figure 14 Blog System Blog Page.....	41
Figure 15 Duration Domain Dataset Overview	46
Figure 16 Frequency Domain Dataset Overview	47
Figure 17 Blog System Use Case Diagram	48
Figure 18 Duration Domain SSE as the Function of Number of Clusters.....	50
Figure 19 Duration Domain K-means Clustering Use Case Heatmap	52
Figure 20 Duration Domain Hierarchical Clustering Dendrogram	55
Figure 21 Duration Domain Hierarchical Clustering Use Case Heatmap.....	57
Figure 22 Frequency Domain SSE as the Function of Number of Clusters	59
Figure 23 Frequency Domain K-means Clustering Use Case Heatmap.....	61
Figure 24 Frequency Domain Hierarchical Clustering Dendrogram.....	64
Figure 25 Frequency Domain Hierarchical Clustering Use Case Heatmap.....	66

Figure 26 Dendrogram of the Subsequent User Type Discovery	68
Figure 27 Frequency Domain Hierarchical Clustering for Subtypes Use Case Heatmap	70
Figure 28 Duration Domain User Type Distribution Comparison.....	75
Figure 29 Frequency Domain User Type Distribution Comparison	80
Figure 30 Hierarchical Clustering for Subtypes User Type Distribution Comparison	84
Figure 31 Interactive User Profiling System User Interface Design	91
Figure 32 Interactive User Profiling System User Interface Design with Subsequent User Types	92

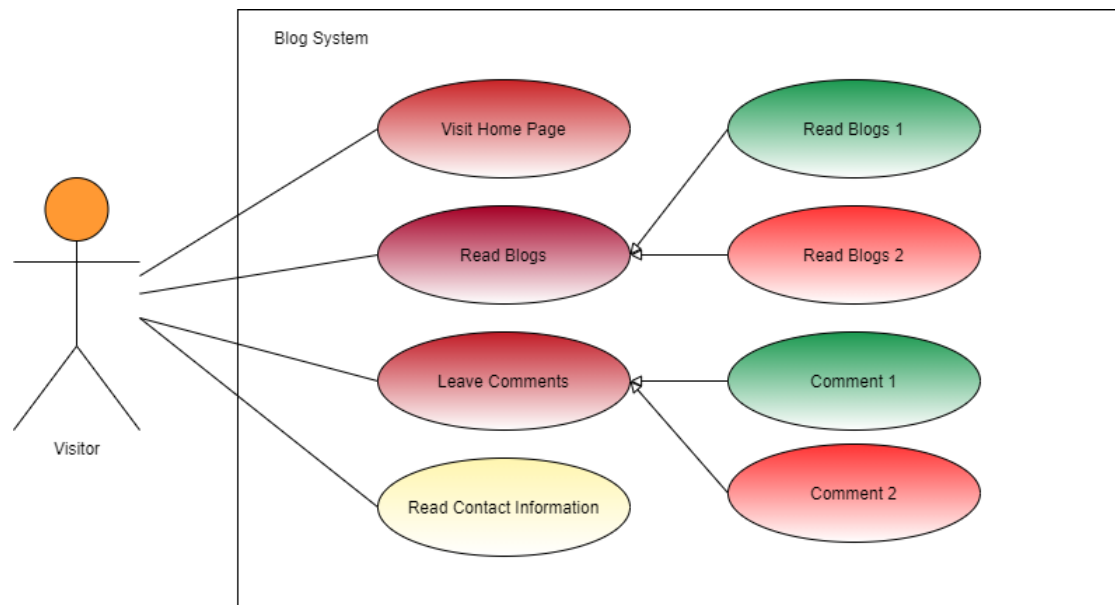
11 Appendix

A. Infographic Use Case Diagrams in Duration Domain K-means Clustering

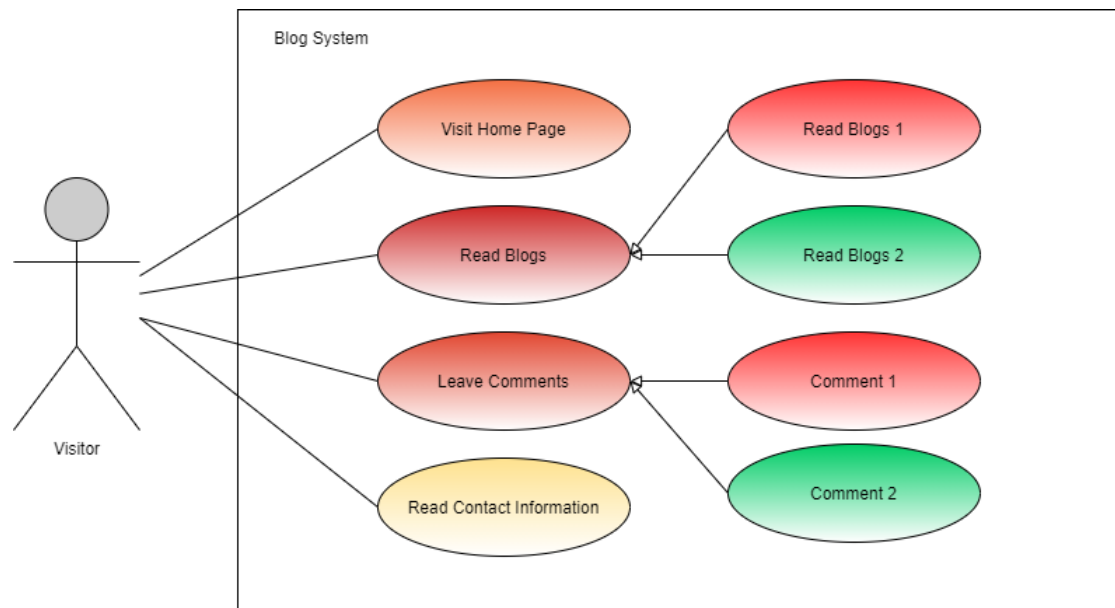
User Type 1



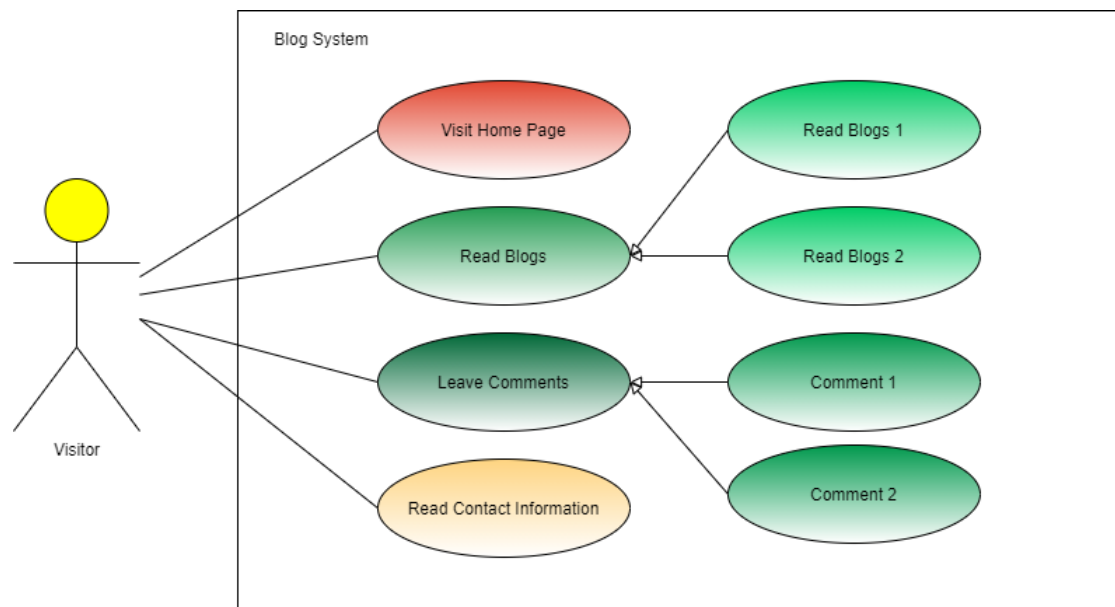
User Type 2



User Type 3

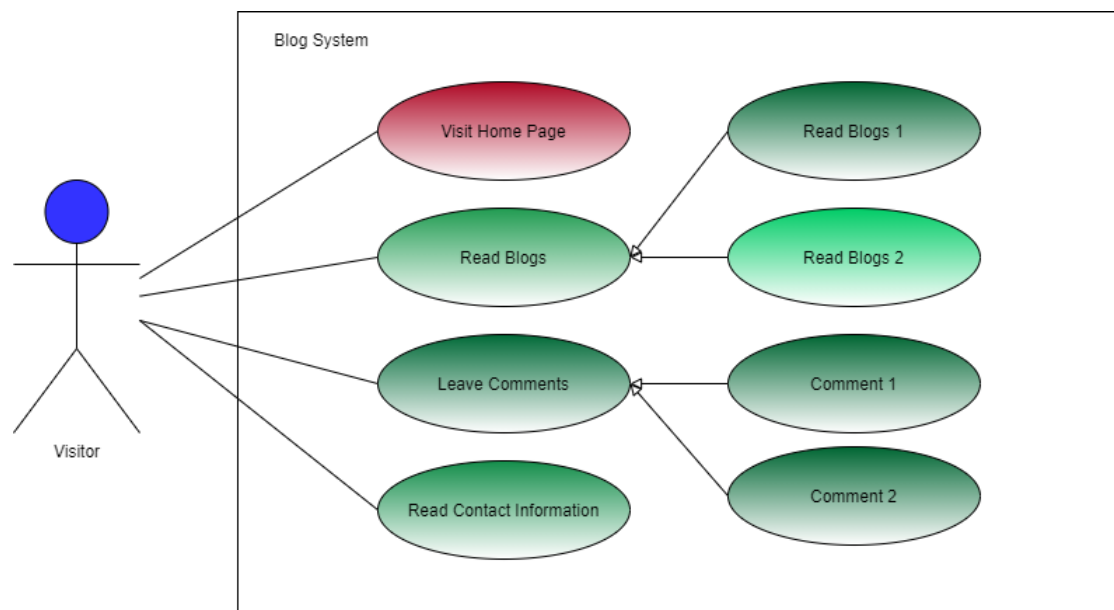


User Type 4

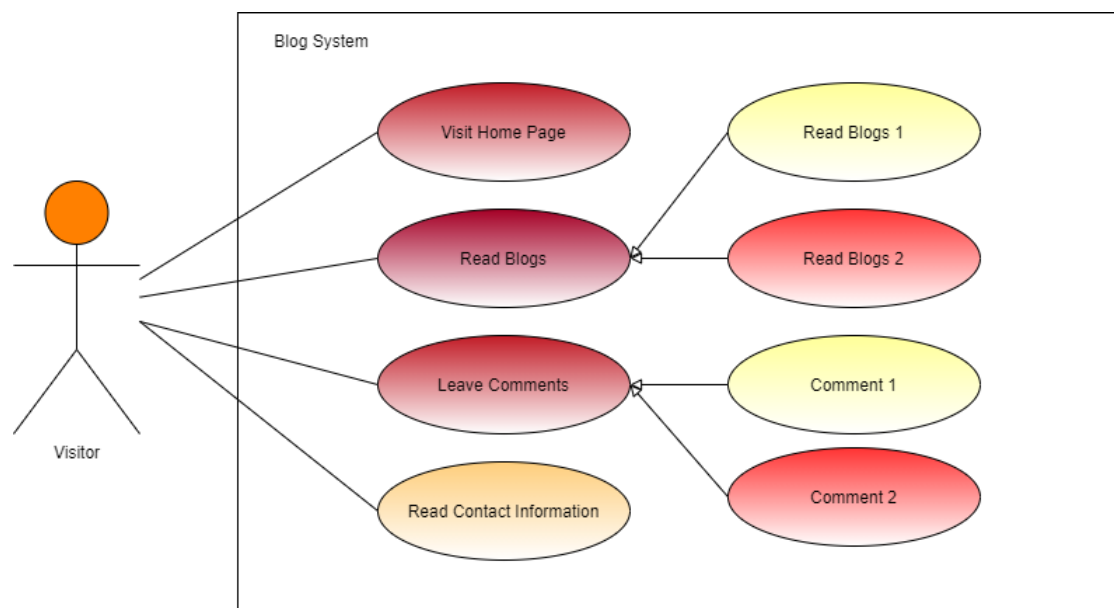


B. Infographic Use Case Diagrams in Duration Domain Hierarchical Clustering

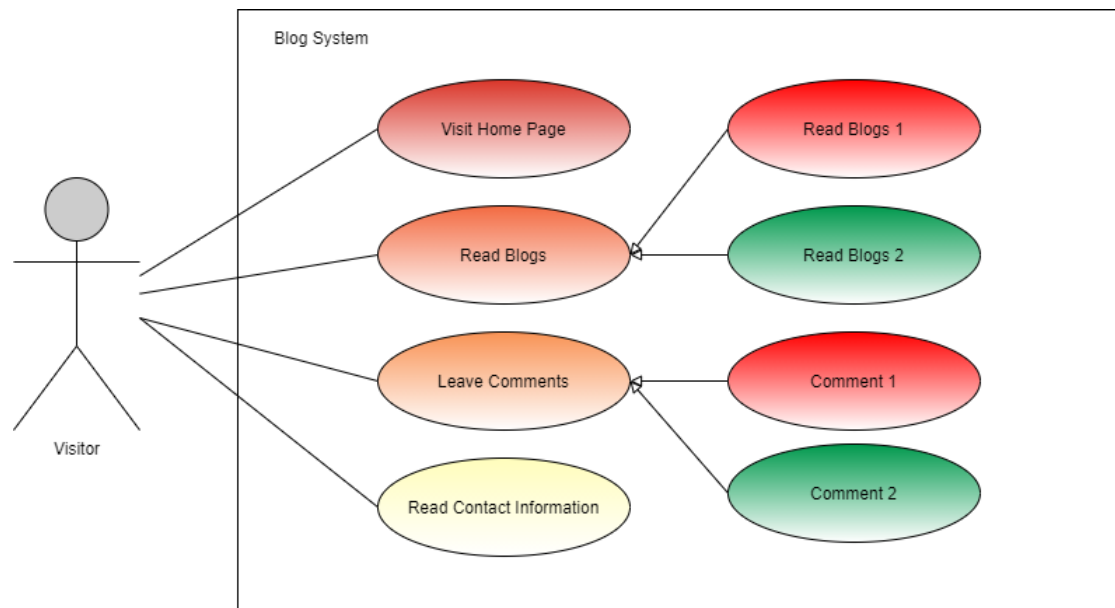
User Type 1



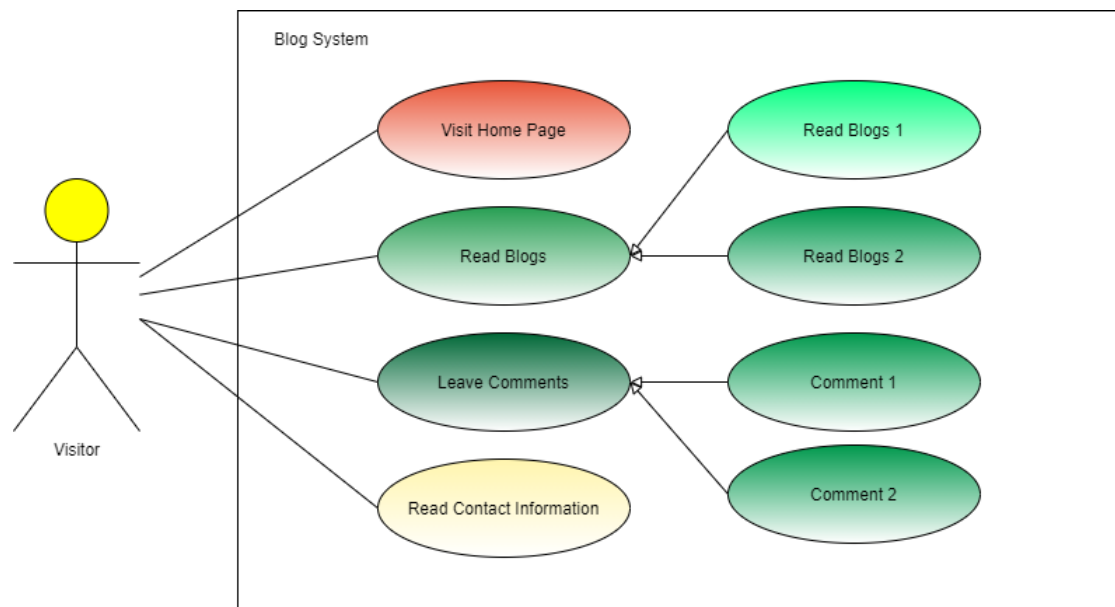
User Type 2



User Type 3

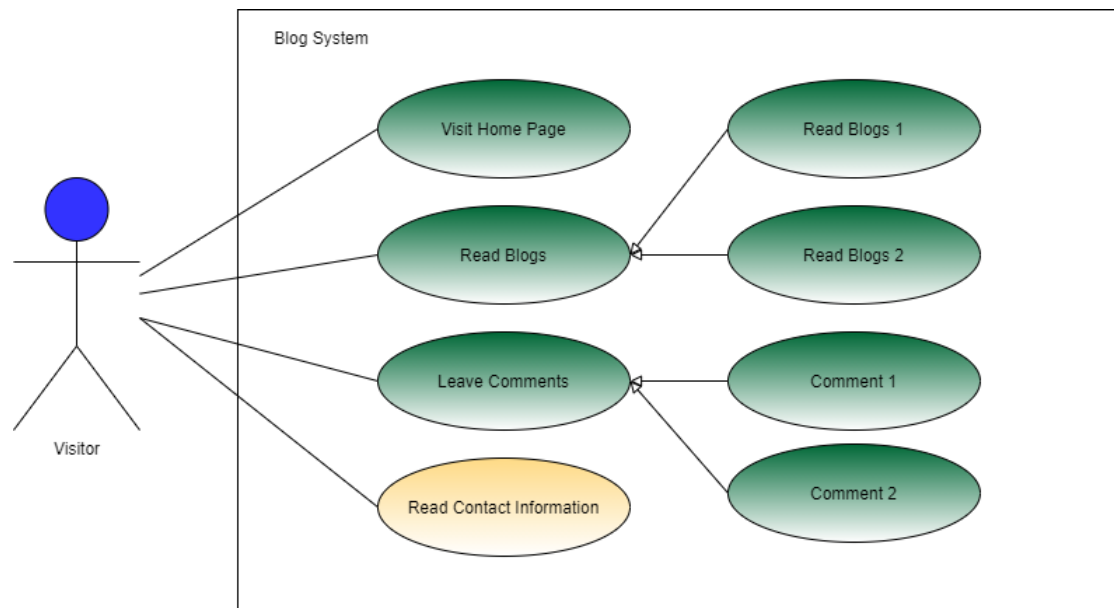


User Type 4

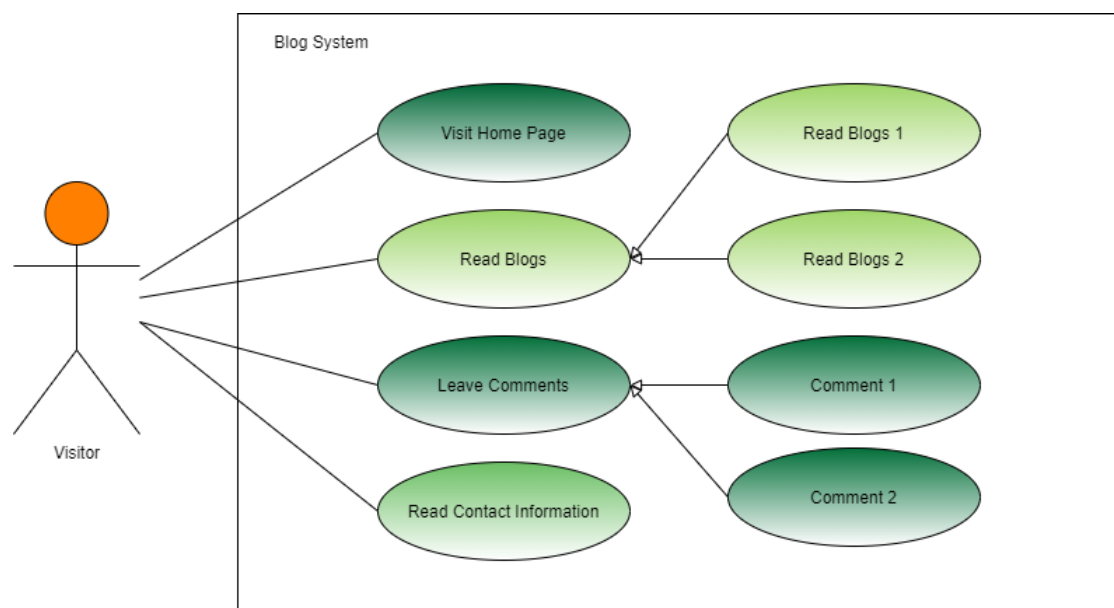


C Infographic Use Case Diagrams in Frequency Domain K-means Clustering

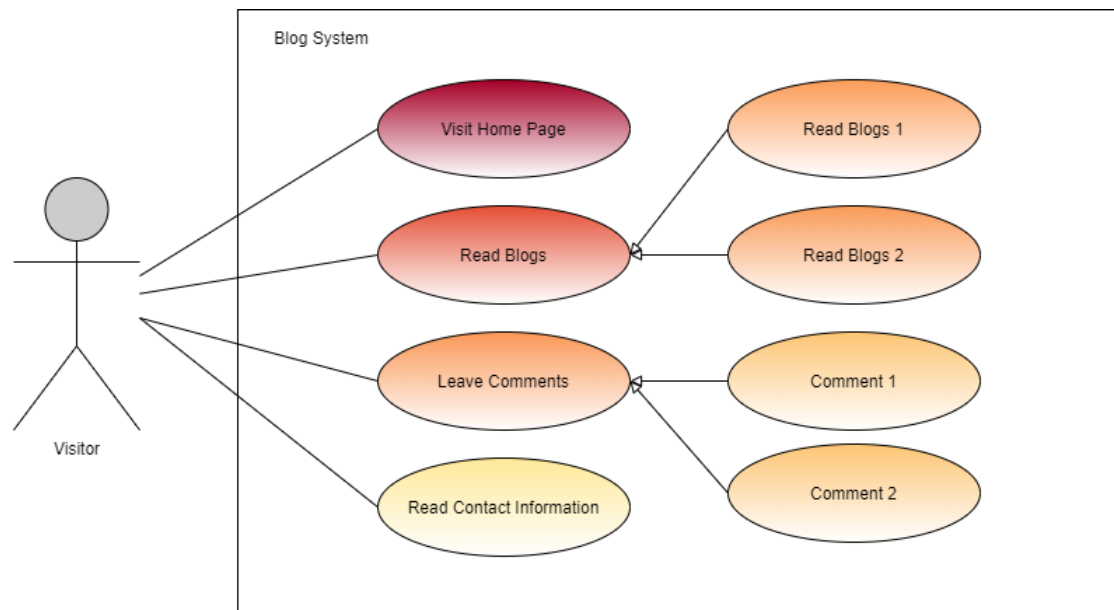
User Type 1



User Type 2

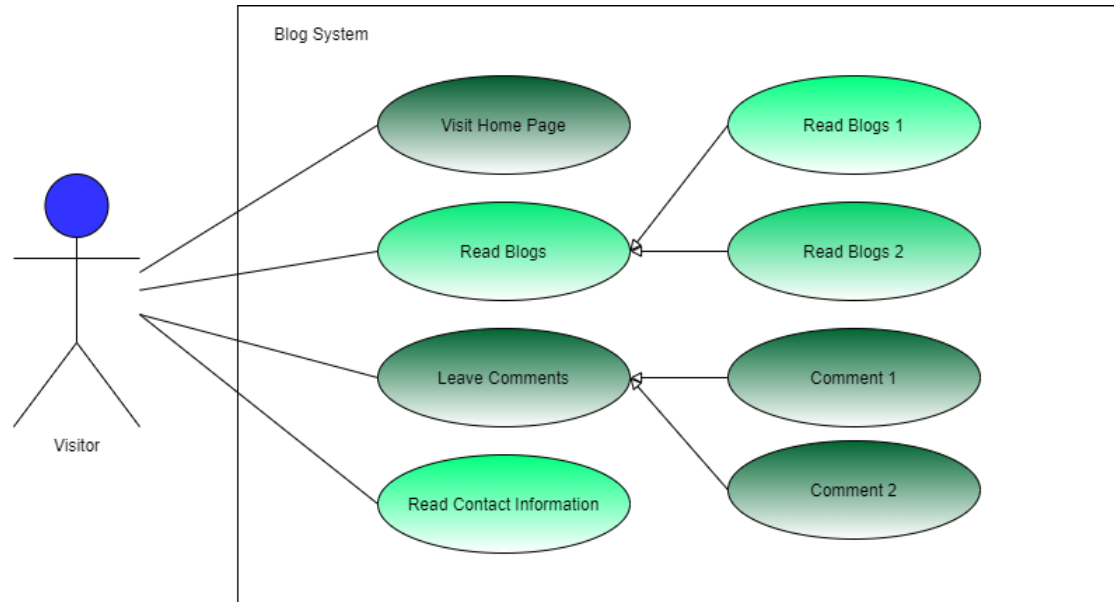


User Type 3

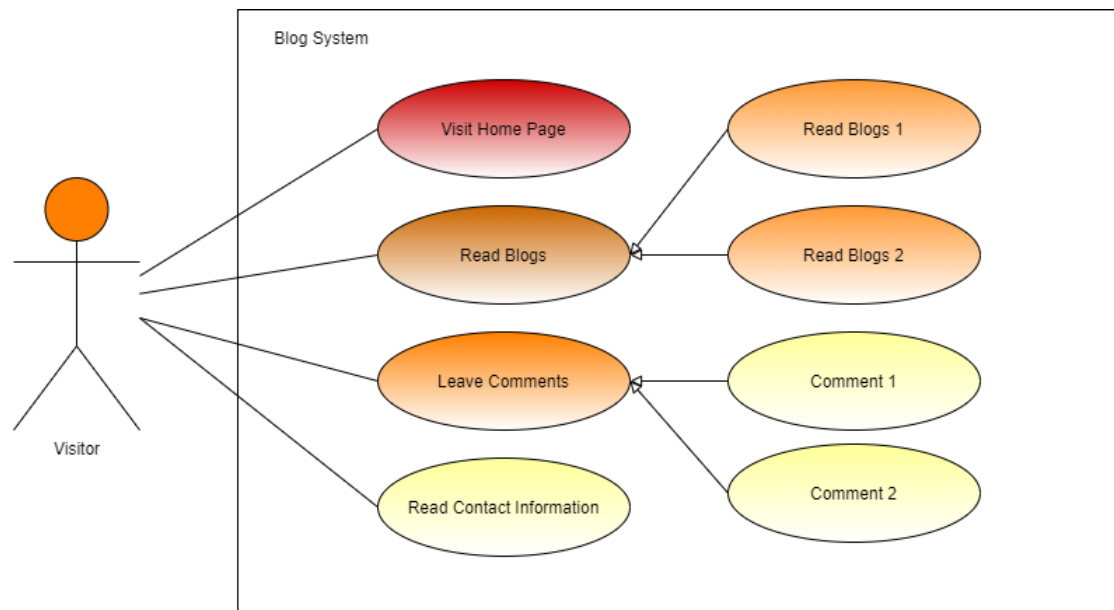


D Infographic Use Case Diagrams in Frequency Domain Hierarchical Clustering

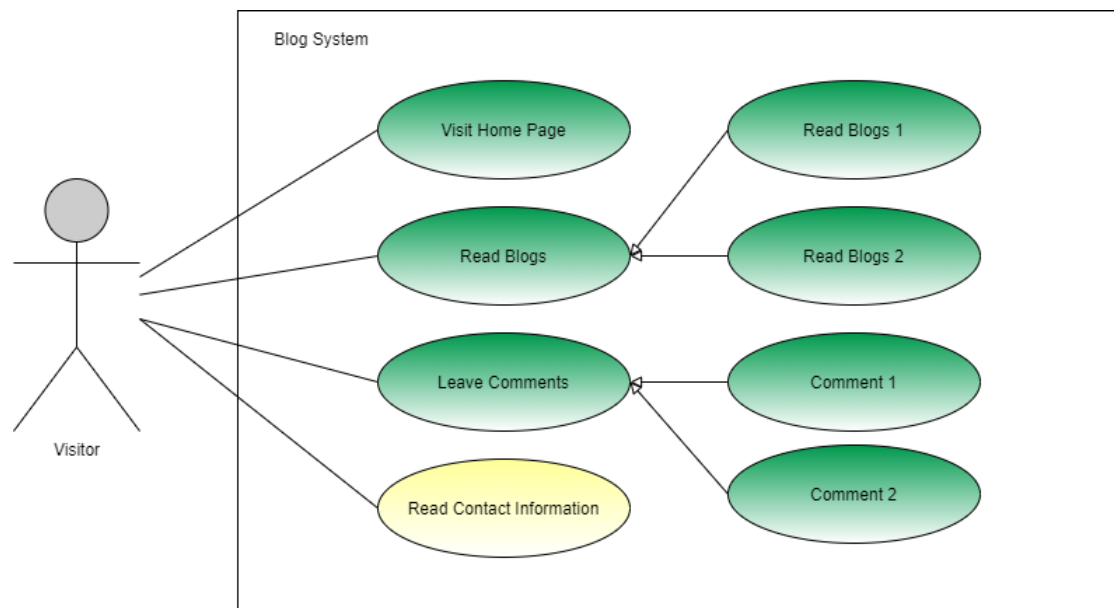
User Type 1



User Type 2

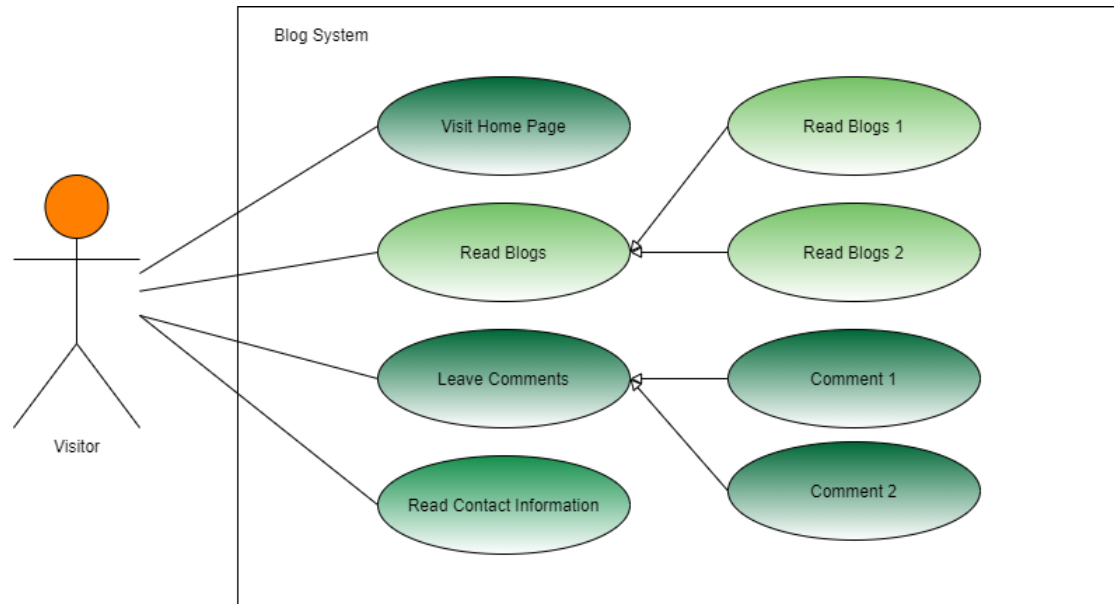


User Type 3

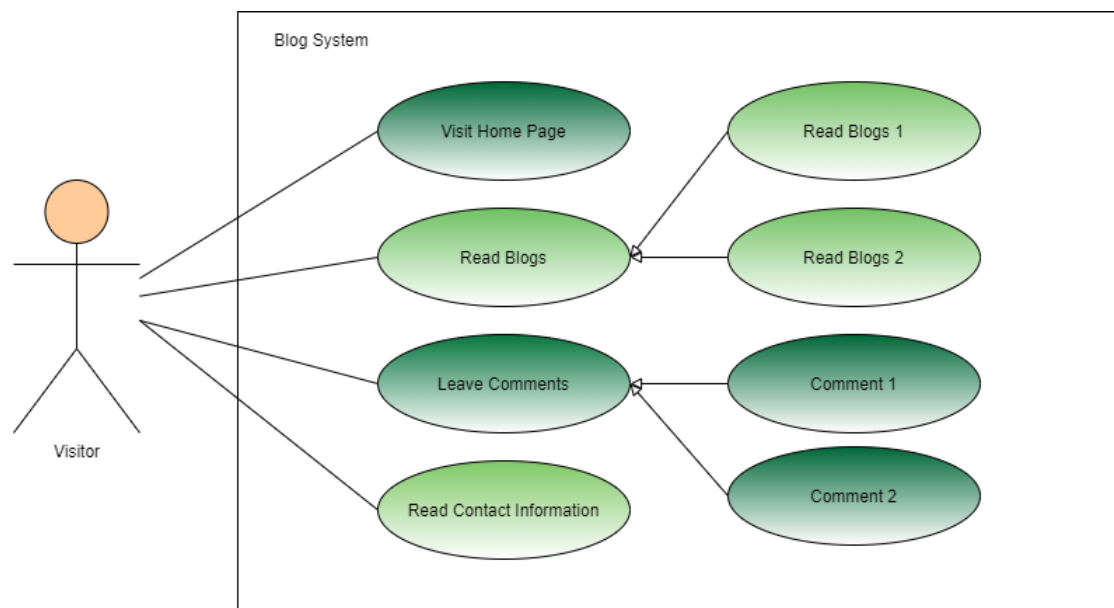


E Infographic Use Case Diagrams in Frequency Domain Hierarchical Clustering with Subtype Discovering

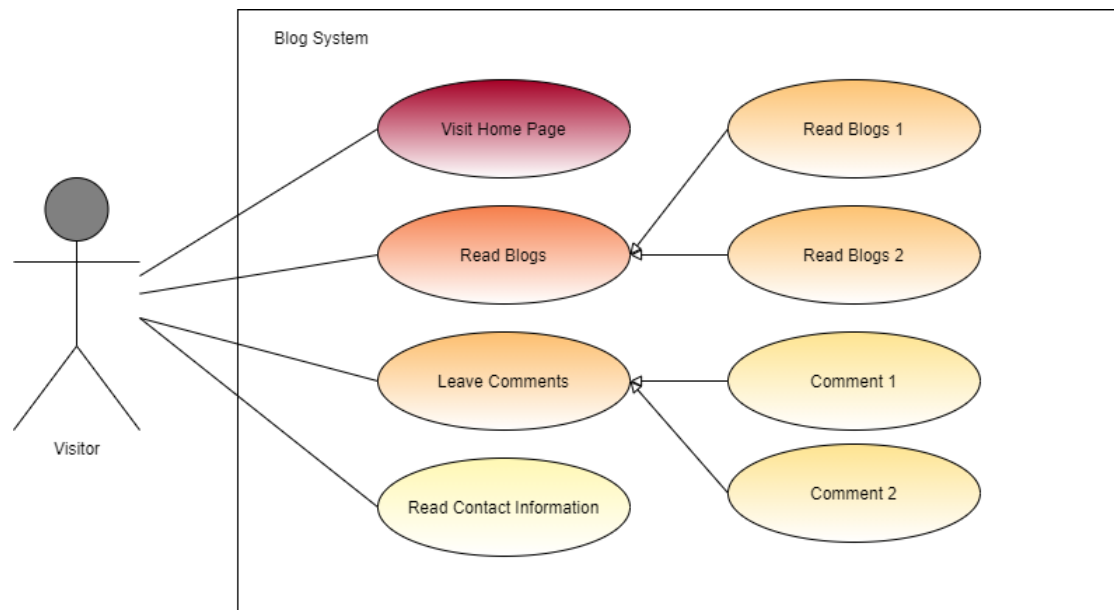
Subsequent User Type 2A



Subsequent User Type 2B



Subsequent User Type 3A



Subsequent User Type 3B

