# Declaration

# Abstract

# Acknowledgement

# Table of Contents

# List of Figures

# List of Tables

# List of Glossary

# Chapter 01: Introduction

## Chapter Overview

In this research project, the author tries to make a Salary recommendation system. This system involves solving a few recommendations system problems and solving salary discrimination problems. This chapter contains information about the proposed project that consists of the problem, aims and objectives of the research, research gap and challenges the author will be facing upon completion of this project.

## Problem Domain

### 1.2.1 Recommendation System

Due to the increase in the use of internet today, there is a huge amount of information accessible to everyone, that also makes it challenging for users to take away essential information provided (Fayyaz et al., 2020). With the over-loaded information and to make the best choice of things that to be shown to the user the recommendation system has helped to overcome this problem (Fayyaz et al., 2020).

There is main three types of recommendation system which are Collaborating Filtering (CF), Content-Based Filtering (CBF) and Hybrid based. Each type of system has its own advantages and disadvantages (Ko et al., 2022).

CF works based on the historical data; this technique tries to forecast the details to a specific user on the things that previously evaluated by the fellow user. This technique relies on memory if the user interactions (Kumar et al., 2022).

CBF technique used to recommend items based on the information the user has done in the past. CBF is the most basic model within the whole recommendation system, that has been used from an earlier stage. (Ko et al., 2022)

However, both kinds of filtering methods had their own limitations therefore hybrid recommendation system was introduced to overcome the limitation in the recommendation system and improve the performance of the recommendation systems.

### 1.2.2 Salary Discrimination

Salary is one of the basic needs of a human being. Salary is also a motivation factor for an employee to work hard and to earn more, since at the end of the day, people study hard and earn much so that they can live a very happy life (Neil Kokemuller, 2017). Hence the salary that is been given by the Companies that the employees work should pay the equal amount for the year of experience the individual has. It is because giving different rates for different individuals with the same performance is a demotivation factor for an individual. Moreover, the individuals are also being paid the same amount as the previous company paid, which is also unfair (Vandenberg, 2020).

In recent times individuals find it difficult to overcome challenges to find a job with the proper appropriate salary (Dr. Jayashree Rahul Pansare, 2022).

## 1.3 Problem Definition

Currently, there is huge confusion on how much amount should an employee gain or should request, when an individual is going to start a job, or while being in the job request for a higher pay. This problem will make the individual quote a less amount during his interview period or demanding period for a higher salary, or individual will be paid based on the previous work has paid. Therefore, the individual will be making his whole effort for the amount that is less paid, which could also be unfair.

As a result, to solve this problem the salary recommendation system will help the individual know how much amount the individual demand should for during his interview and be confident and bargain for the salary suggested or bargain for more than it is suggested. Due to this, it will help the individuals to overcome salary discriminations, since individuals know the exact amount that they will have to be paid according to the individuals experience in the IT industry.

### 1.3.1 Problem Statement

Earning the appropriate salary for effort that an individual does is very important. Therefore, when an individual is trying to switch to another company or trying to request for a higher salary, the individual should know the figure to bargain, else individual will be quoting less than the expected amount since individual has no prior knowledge.

## Research Motivation

According to the problem defined due to discrimination of salary in between the individuals and the individuals are finding it difficult to find appropriate salary for the job (Dr. Jayashree Rahul Pansare, 2022). Thereby the author believes in developing a recommendation system for the people around the IT industry to know about their best fit salary for the year of experience that the individuals have. This indeed will help the individual to bargain or demand the salary that an individual is intended to gain.

## 1.5 Similar works

|  |  |  |  |
| --- | --- | --- | --- |
| Citation | Technique Used | Improvement | Limitations |
| (Tiwari et al., 2022) | This thesis helps to recommend the best lawyer based on the user request for. Based on their income, area and so on. This system has used a hybrid model by using the algorithm AGE- MOEA (Adaptive Geometry Estimation based Multi Optimization  Estimation  Algorithm) | The system has made a hybrid recommendation model that has helped to improve the accuracy of the recommendation system. The AGE- MOEA has the accuracy of 99.12% | The lawyer recommendation system is solely made for Indian cases and Indian lawyers. |
| (Sharma, Rana and  Malhotra, 2022) | This is an approach to recommend to the user what type of products they need based on the user actions. This system has used all their kinds of approaches to do the recommendation. And have made their solution using the hybrid recommendation system using both Collaborative Filtering (CF) and (Content Based Filtering) CBF to give them a better recommendation for the user. | The improvement in this system is that the accuracy, which has been evaluated by the Precision, recall and F measure metrics | The future enhancement that the system wanted to use the machine learning model such as the support vector machine and naïve baiyes. Also, to check if their accuracy is good by using it with real user acceptance level. |
| (Kumar et al., 2022) | This system has made to recommend job to the users by using hybrid recommendation approach with the help of web scraping and web crawl the application was able to extract data to make the best fit job to the user that wants to seek job in the industry that intended to work. | By the implementation of this system, it helped to overcome the limitations. Also, the system was increasing the efficiency ranking as well. Moreover, problems such as cold start, sparse database scalability and lack of trend recommendation have been eliminated. | Automating the web crawler.  Implementing the accuracyP in content-based-filtering will improve the accurate match of the search.  Improve the collaborating filtering by giving different weights to different users based on their LinkedIn Skills. |
| (Singh, Sabharwal and Gabrani, 2022) | This research paper is trying to implement a recommendation system by using a MovieLens dataset to check if the grasshopper algorithm fits well into a Collaborating Filter recommendation system. | The system was not able to improve the accuracy, but it was able to state that grasshopper algorithm could also be used in the collaborating filtering since it was able to give a decent rank value of **6.14E+04** | Using the exponential Grasshopper algorithm for multi-objective problems. This algorithm could be used in tried using in other domains such as image segmentation big data, and wireless sensor networks. Etc. |
| (S et al., 2022) | According to the research that has taken place the user is being suggested for cosmetic products that are being able to be used by their own skin sensitivity. Therefore, this system has used web scrapping to get the information of the chemicals that has been used in the products. | By using natural processing language (NLP) and Machine Learning (ML) the system was able to generate an accuracy of 93% with the dataset that they have used. With the external dataset it has been 85%. | The future enhancement is to train the system in way to Analyze for any newly added product.  To improve the existing algorithm to increase the accuracy of the system.  Include more parameters such as: - price, standard, rank etc. |

Table 1 Similar Works

## 

## 1.6 Research Gap

In fact, based on the similar work that was stated related to recommendation system, was on domain such as career, job, movie etc. But this thesis will be doing a recommendation system for Salary. Which means that it will recommend the user that uses the system about the amount of salary that the user will be expecting to get based on the year of experience that the user has stated. Therefore, this project will make sure to resolve the issues in recommendation system such as cold start, Data sparsity etc. (Fayyaz et al., 2020). and social problem such as Salary discrimination(Quan, 2020), in the Salary domain either my using CF, or CBF or by implementing a hybrid recommendation model.

## 1.7 Contribution to the body of knowledge

### 1.7.1 Technology Contribution

A hybrid recommendation system will be used along with the attempted domain that is salary. This hybrid technique will try to make sure of different machine learning techniques to improve the problems in the recommendation system.

### 1.7.2 Domain Contribution

Due to overloaded of information throughout the internet (Gohel and Vanjara, 2022), it makes people difficult to get to know the suitable things that they should acquire for themselves. As per this thesis, it has been considered as Salary. Salary information is very crucial to acquire, and it is very difficult for an individual to get to know the level of cash should an individual be paid for the effort or experience an individual has. Hence the recommendation system in the Salary domain will be a new attempt. Which will recommend salary for the user based on their experience level. This intern will help the domain to remove the discrimination of salary that has been established between the sex group (Quan, 2020b) and boost up confidence for an individual to bargain on the salary that it’s been recommended.

## 1.8 Research Challenge

There could be couple of challenges that will be faced while implementing this approach, since there is not any research happened to recommend salary, which is fully on numeric values, it is challenging that how to implement a better hybrid recommendation system for the hypothesis.

In addition, selection of a better algorithm or technology will be very crucial since the recommendation should be more appropriate for the details that the user needs based on the level of expectation that an individual needs, which intern will help an individual to bargain for their payment whilst the individual is in the job or trying to apply a new job in different company. It is very challenging because the relevant factors have been considered whilst doing the model, which will give the highest and valuable information to the user.

## 1.9 Research Questions

**RQ01**: what are the current algorithms that have been used to build a hybrid recommendation system?

**RQ02**: how has recommendation system helped to improve the suggestions for the users?

**RQ03**: what are the factors that a company decides to pay wages for an employee?

## 1.10 Research Aim

The aim of this research is to design, develop and evaluate a Recommendation Architecture that will help the individuals to know the amount of salary to be bargained during that switch to a new job, or higher position promotion, else the individual could be made discriminated or biased.

This hypothesis’ aim is to build system and architecture a recommendation system with the aid of specific data set that was used from the relevant source. The focus of this research is to recommend salary amount that an individual is eligible to get based on his prior experiences. To make a better recommendation system for the salary domain that has never been done before the research will be conducted on Data Mining, CF, CBF, hybrid system, hence will lead to a better hypothesis for the public to check on the salary expectation that individuals would gain.

Moreover, since there is not any research paper that has taken place in this domain, it’s possible to publish a research paper in the finding of these project solutions. The knowledge to implement these ML/DL models will be researched and studied to validate how far has this hypothesis could help the public sector people in the IT industry to get a better knowledge of their salary incomes. Furthermore, the other aim of this hypothesis is to that the system will be hosted in the public server, which intern will help the users for further enhancement in their career.

## 1.11 Research Objectives

As according to the Aims and Objective mentioned above, are those of that will be achieved and answered with the completion of the following research objective. These objectives will be reached to complete this research successfully and feasible to be used by the public.

|  |  |  |
| --- | --- | --- |
| Objective | Description | Learning Outcome |
| Problem Definition | The author identified a research problem in the domains based on the author preference and interest and analyze the problem on how to validate the desired problem identified. | LO1 |
| Literature Survey | To make a successful literature survey, should read previous work that is relevant to the thesis and critically evaluate them.  **RO1:** Systemically should analyze on what are the type of existing work that has taken place, or similar domain that already been researched.  **RO2**: To identify the algorithms, technologies and techniques used to make a recommendation system.  **RO3:** Study the background of recommendation system its pros and cons and how it has helped the world to improve.  **RO4:** Background study about the Salary discrimination and it has affected the employees. | LO1, LO4 |
| Requirement Elicitation | Gathering information about the gaps and domain related requirements.  **RO1:** Gathering requirements from the users, on how the recommendation system that individuals are expecting.  **RO2**: Getting feedback from the domain experts and prior users before deploying the system. Example: Look and feel of the system etc. | LO2, LO3, LO5 |
| Design | Designing the proposed system architecture  **RO1:** Design a Model with the relevant dataset available to build the foreground of the recommendation system.  **RO2**: Design a backend server that will allow requests to flow in and recommend the relevant specific things to the end user.  **RO3:** Design a GUI for this system, that allows user to interact with. | LO1, LO5 |
| Development | Developing a prototype using relevant software and hardware resources to make the system.  **RO1:** Develop the model for the salary recommendation system, with the dataset that has been found.  **RO2:** Develop front-end system, that helps the users to interact with.  **RO3:** Develop a backend server.  **RO4:** Deploying the proposed system to the cloud environment. | LO1, LO5, LO6 |
| Testing and Evaluation | Testing and evaluating the test cases is a key to any building software since that gives evidence that the system is working as expected.  **RO1:** Analyzing and doing the black box and white box testing.  **RO2:** Get evaluations from the users and other domain experts regarding the system.  **RO3:** Evaluating on how well this system has helped the users to get to know a better salary recommendation. | LO8, LO9 |
| Documenting the progress of the research | Forecasting and documenting the progress of the research project. | LO6, LO8 |
| Publish Findings | Doing well documented, structured report and critically evaluating the research.  **RO1:** Publishing a research paper since there is not any paper published in this domain.  **RO2:** Publishing the codeor idea to the public hence individual could check on how the implementation has taken place. | LO4, LO8 |

Table 2 Research Objectives

## 1.12 Project Scope

The main aim of this study is to make a well architecture Salary recommendation System for the employees, using this system to know about the expectation of their salary. As per the objectives and aim of this thesis, the scope of the requirements is mentioned below.

### 1.12.1 In Scope

* Building proper architecture recommendation model by using ML/DL algorithms.
* Implementing a fully-fledged application for the User to interact with the models and suggestion that individual expect.
* Evaluating how well this system has helped the employee around based on the individual’s necessity.
* Testing the system by checking on the accuracy, the precision score, and other possible metrics.

### 1.12.2 Out Scope

* These will be the parts that will not be covered in this research.
* The system will be limited to the IT Industry only.
* The system dataset has been trained purely based on Sri Lanka salary rate, hence it will not for the worldwide use.
* Only a few input features will be taken into consideration such as the Designation, Work experience, Company size, Year of experience.

### 1.12.3 Prototype Feature Diagram

### Diagram Description automatically generated

Figure 1 Prototype

## 1.13 Chapter Summary

The research gap, the research challenge, and the research strategy that the author is expected to solve in the research project offered by this document were all stated in this chapter together with the appropriate proofs and domain descriptions. The study's goals were to link to the project module's learning objectives in the University of Westminster's BEng Software Engineering undergraduate program.

# Chapter 02: Literature Review

## 2.1 Chapter Overview

According to the previous chapter the author discussed about the problem domain that the research will be based on, In this chapter the author will be explaining in dept about the problem with evidence on how have the similar or existing works have been taken place to solve the issues, hence author will be skim through the techniques and algorithm already been done to evaluate the issues in the problem domain.

## A picture containing drawing, diagram, text, sketch Description automatically generated2.2 Concept Map

Figure 2 Concept Map

## 2.3 Salary Discrimination

Even though over 25 years of time the gender discrimination for salary payment is narrowing down little by little, according to research still their gender gap in the IT industry regarding to the enumeration that an individual receives (Quan, 2020). It is not only the IT industry that is being affected by the pay gap, but many workplaces that has gender discrimination have led to this kind of pay gap and also being the prominent topic worldwide, the remuneration is not being differed by the employee working hours or their qualifications but also by gender (Khatri, 2022). Salary is one of the most important for an individual due to efficient wages being paid is a motivation to optimal production and informative of employee ability, as a result the turnover in the labor market will be reduce if this wage discrimination is turned out.(Meli and Spindler, 2021).

Moreover, salary discrimination is not based on the skills or the education level an individual have, the main other factor that leads to discrimination in remuneration payment is the Region or based on the different location, according to the location that an individual is in the remuneration differs from another location, since the standard of living will be totally different. (Zhou, Bu and Gao, 2021).

According to the study, in between 2018 to 2019 women I the tech career is being offered less than men for the same job, and sometimes in the same company which has led up to 63%. As a result, even though the Tech industry believe that the gender pay gap is narrowing, it false since it is growing throughout.(Perry, 2020). Apart from the gender, location above it is also said that the salary history causes individuals to have discriminated pay (Safstrom, 2019), hence each employees of the same job career of same level experience will earning different remuneration.

Inequality of paying remuneration in this current society is not only based on the gender, it is also decided based on the peoples color, LGTBQ communities and other ethnic minorities (Cziesielski, 2020), hence the author has decided to proposed this thesis in order to overcome some kind of salary discrimination in the IT industry based on Sri Lanka salary information.

## 2.4 Approach taken to solve Salary Discrimination

### 2.4.1 Banning of salary history.

One of the ways in which to make salary pay discrimination is using salary history in setting up the wages. Hence during the process of hiring an employee to the organization the employer should avoid inquiring regarding the individual salary history and adopting “reckless discrimination” legal theory could promote greater pay equity in the industry. (Vandenberg, 2020)

## 2.5 Factors considered whilst considering salary payment.

### 2.5.1 Education

Education is an investment according to the human capital therapist. It has been demonstrated that education increases a person's capacity for production, employability, and discretionary income, which in turn lowers the cost of community social services and creates funds to fund additional education. Additionally, it has been discovered that education raises a person's quality of life by giving them more free time and resources to enjoy it (Owings and Kaplan, 2019). Due to higher education with a better knowledge of the domain that an individual will be earning more than another individual since reducing the risk of loss will be minimal according to the education level an individual holds to (Quan, 2020b).

Moreover it clearly indicates that when an individual has a better education qualification, the individual has a higher chance of obtaining a higher salary.(Wang et al., 2022)

### 2.5.2 Experience

Experience is another human capital factor. By having a higher experience it during on-job training will enable an individual to acquire a versatile skills which will be an added advantage since the knowledge in the domain will be very much higher than the new fresh graduate or an individual who is in his intern period (Quan, 2020b).

### 2.5.3 Type of work selection

The remuneration that will be paid to an individual will also be considered based on the type of working selection selected. The possible working selection could be either temporary or permanent contract (Marin-Garcia and Martínez-Tomás, 2022), as a result the salary payment for an individual will also be different apart from the gender discrimination.

### 2.5.4 Geographical Location

The remuneration that will be given to an individual will also be affected by the geographical location they are working on, since the economy will be different compared to other countries across which could be either developing, developed or poverty state, as a result the IT industry individual will be given different remuneration even when they working on the same job role across different region (Zhang and Chung, 2018).

### 2.5.5 Environment of the Industry

As per the above factors the remuneration that could be earned by an individual is based on how well the company has been established in the industry (Wang et al., 2022). The higher it has been established the company tends to make higher profit due more projects being involved compared to start up, as a result the salary will be differed.

## 2.6 Recommendation System

Due to increase in information over the internet it is very much difficult for users to obtain data that is most useful for them, therefore recommendation system has been increasingly popular model used to give the best suggestion according to the user preferences (Dhinakaran et al., 2022). The development of smart devices has caused a huge traffic in web, App and SNS platforms, which collects various information related to the users on whatever the user uploads or share among people across the internet, hence the recommendation system should well use these data to make the better preference as well as perform in a better way for the users across the globe (Ko et al., 2022).

## 2.7 Problems in Recommendation System

Even though the recommendation system has its own advantages, it has different problems when building a recommender system.

### 2.7.1 Cold start problem

Cold start problem arises when there is no information found about the user or item in the system, since the user has just started to use the application. An example can be when a user joins Netflix. The second problem that is possible to arise is when there is a new item that has been added to the system the collaborative filter technique uses user-item matrix to give recommendation, since the new item is not given any rating it is unable to give proper recommendation for this item which is known as cold start problem.(Mazumdar, Patra and Babu, 2020)

### 2.7.2 Sparsity problem

Sparsity problem arises when a user gives false information about the rating feedback that user gives to the product that has been recommended to them, since rating seems useless for the users (Mishra et al., 2021). Without realizing the user might give a higher rating considering its best for them or lower rating considered non likely without even noticing the type of the product. These ratings are taken as an input to the recommender system which in turn might display unwanted results to the end user, indicate loss of interest for the user in the platform and lead to non-efficient working (Kitazawa and Yui, 2018). Due to sparsity of rating matrix this kind of problem will arise.

### 2.7.3 Scalability problem

When a system is published there is a possibility the number of users could be growing up day by day, therefore will the system be able to cope up with the same performance level, since when the user of the system increases it will face scalability issue it become slow and make it feel it start to give problem in the recommender system, could e due to hardware and software scalability (Wang and Ke, 2014).

Moreover, the accuracy of the prediction could also be redundant or reduce due to increase in data, some of the algorithm are not being able to take advantage of increased efficiency of the hardware, which creates a problem of scalability (Da Costa et al., 2018).

### 2.7.4 Lack of Data problem

The efficiency of recommendations algorithm is evaluated by using publicly available dataset from different environment has been a common practice to make a recommender system, but recommender system faces a big issue to make an effective recommendation due to lack of data (Zhang et al., 2010). To make a better recommendation in any environment if there are more consumer/item data it would help to build a better recommender algorithm(Lika, Kolomvatsos and Hadjiefthymiades, 2014).

In addition to all the problems that have been specified above, there are also few problems such as Changing data, changing user preferences, unpredictable items which are also considered as a problem but based on the proposed system the salary recommendation system, this problem will now be suitable.

### 2.7.5 Over Specialization problem

This recommender problem arises when the recommended items are too like each other. When the user keeps on seeing the same item over and over again and not any new unique data, it might make the user loose interest on using that website since, the same result is showing over the time (Adamopoulos and Tuzhilin, 2014).

## 2.8 How will the recommender system help the Users?

### 2.8.1 Personalization

By giving users personalized recommendations based on their tastes and behavior, recommender systems can become more personalized. Collaborative filtering, content-based filtering, or hybrid strategies can all be used to accomplish this. Personalization is a crucial component that raises customer pleasure, loyalty, and engagement (Gorgoglione, Panniello and Tuzhilin, 2019).

### 2.8.2 Time saving

Reducing the time and effort needed to obtain pertinent information or items is one advantage that recommender systems can offer users. Users can quickly and easily identify new products or material that they are likely to appreciate or find useful by automatically producing individualized suggestions, without having to spend a lot of time searching or scrolling through extensive catalogs or websites.

In today's fast-paced, information-rich world, when users are frequently overwhelmed by the sheer volume of available information and content, this time-saving benefit is especially crucial. User-friendly recommender systems can assist users in quickly navigating this information environment and locating what they need or want (Wairkar et al., 2021).

### 2.8.3 Improved decision making

The advantage of offering customers individualized recommendations based on their interests and behavior is referred to as improved decision making in recommender systems. This aids users in avoiding decision fatigue, overcoming information overload, and making better decisions(Forouzandeh, Rostami and Berahmand, 2022).

## 2.9 Similar Works

### 2.9.1 Classification Recommendation system

Based on the research it has been clearly stated that recommendation system can be improved using other machine learning model such as Decision tree, Random Forest to make recommendation than not only depending on content-based filtering and collaborative-filtering (Bhansali and Nagwani, 2021).

**Multi-Domain Recommendation System Using Hybrid Filtering and Support Vector Machine Classification** (Vasanth, PeriyaKaruppan and PoornaKumar, 2020)has proposed a hybrid recommendation system which combines collaborative, content-based as well as support vector in order to improve the recommendation accuracy. The main domain that has been selected by the author is the movie domain. The process of the system first uses the collaborative and content-based filtering to generate recommendation based on the user-item and item features, based on past interactions and characteristics of the user and the recommended products, the system employs a support vector machine classifier to predict whether the user will like or dislike the recommended items. The experimental results indicate that by using a hybrid approach the overall recommendation quality will be improved and overcome the problems the recommender systems.

**Classification Algorithm for Career Recommendation System**(Masika, Rono and Kati, 2022)**.** The study suggests a career recommendation system that matches job searchers with appropriate employment roles using categorization algorithms. The method considers several variables, including the education, experience, talents, and interests of the job seeker as well as the credentials, job type, and geographical criteria.

In terms of recommendation accuracy, the authors evaluate the performance of three different categorization algorithms: decision trees, random forests, and support vector machines (SVM). They make use of a dataset that contains the profiles of job searchers and job specifications, and they assess the algorithms' performance using a variety of performance metrics, including accuracy, precision, recall, and F1-score.

The experimental findings demonstrate that the SVM algorithm, with an accuracy rate of 93.75%, surpasses the decision tree and random forest algorithms in terms of recommendation accuracy. To assess the system's usability and get user input on the suggested changes, the authors also carry out a user study.

Overall, the article offers a potential method for creating classification algorithms-based career recommendation systems. The imbalance between supply and demand in the labor market can be closed by the suggested system by assisting both employers and job seekers in locating appropriate workers.

### 2.9.2 Content based Recommendation system.

**Solving Cold start Problem for recommendation system using Content Based filtering**(Chia and Najafabadi, 2022)**.** To address the cold start issue in recommendation systems, the study suggests a content-based filtering technique. The lack of user-item interaction data, known as the "cold start" problem, makes it challenging to produce precise recommendations utilizing collaborative filtering approaches. The cold start issue can be solved by content-based filtering, which generates suggestions based on the qualities of things.

The authors employ a vector space model to represent the elements in a dataset of movie properties, including title, genre, and storyline synopsis. The similarity between the user's preferences and the item attributes is then calculated using cosine similarity, and the user is then given customized recommendations.

The experimental results demonstrate that the proposed technique performs better in terms of recommendation accuracy and diversity than a few baseline methods. To assess the system's usability and get user input on the suggested changes, the authors also carry out a user study.

Overall, the research offers a viable method for utilizing content-based filtering to address the cold start issue in recommendation systems. By raising the quality of recommendations for new users and items, the suggested method can improve user engagement and experience.

### 2.9.3 Salary prediction system

**Salary prediction using random forest with fundamental features**(Chen, Mao and Yuan, 2022).The purpose of the study article is to use random forests with basic features to estimate individual incomes. Included among the essential characteristics are education, years of experience, job title, and firm size.

The study makes use of a dataset including details on worker incomes, job descriptions, business sizes, educational backgrounds, and years of experience. The data is preprocessed, feature engineering is done, and a random forest model is used to forecast salaries.

The findings demonstrate that the random forest model with core features is highly accurate in predicting salaries. To ascertain which features have the most bearing on wage projections, the authors also conduct a feature important analysis.

The study offers insightful information on how to forecast salaries utilizing fundamental features using machine learning techniques, specifically random forest. Additionally, it emphasizes how crucial feature engineering and feature importance analysis are for enhancing model performance.

## 2.10 Technology Review

Recommendation systems can be built in different ways by using recommendation techniques or by using machine learning algorithms or with the help of deep learning. Even though this method has been proven to perform well it has its own limitations as well, hence according to the business requirement the recommendation techniques and algorithms should be chosen. Building a proper recommendation system with an effective algorithm based on the requirement would help to improve the sales of a business(Jannach and Jugovac, 2019).

### 2.10.1 Recommendation system techniques

|  |  |
| --- | --- |
| Techniques | Description |
| Content based filtering | Content based filtering algorithm is an Information retrieval Technology (Wu, 2022), which recommends or extract information based on the passed or newly added items that matches the users operation behavior.  Steps to do content-based filtering:   1. Extract the information from the people of events that have taken place. 2. Should transform the feature sets into feature vectors. 3. Use the vector space and then make comparison with Cosine Similarity. |
| Collaborative filtering | Collaborative filtering can be divided into 3 different categories.   1. User based collaborative filtering– This algorithm uses user information which will get the user with many similarities from the organization to that makes mutual recommendation between each other. 2. Item-based filtering – This type of algorithm could be used when the number of items is less compared to the number of users in the system, this would recommend items like the target product that the user has selected. 3. Model based collaborative filtering – This approach is based on the user’s behavior and their preference information.   Please check diagram for more clarification about User based and Item based filtering. Please check the diagram in [**APENDIX A - User and Item based filtering Diagram**](#_APENDIX_A_-) |
| Knowledge based filtering | Knowledge based recommendation will be using the knowledge about the users as well as the product that the user is interest in, as a result the product that is recommended will be based on the user’s requirement (Burke, 2000). |
| Hybrid recommender system | This type of recommendation system is a combination of two recommendation model, or it could be different kind of machine learning algorithm that is combined with one of the recommendation techniques. |

Table 3 Recommendation System Technique

### 2.10.2 Machine Learning algorithm for recommender systems

|  |  |
| --- | --- |
| Models | Description |
| Random Forest Classifier | The Random Forest algorithm is a supervised learning system that uses decision trees as its foundation. To arrive at a final forecast, it builds several decision trees throughout training. The algorithm grows each decision tree using a random subset of the available features and data samples. This randomness enhances the algorithm's capacity for generalization and guards against overfitting (Shingade, Mudhalwadkar and Masal, 2022). |
| Decision Tree | The Decision Tree algorithm is a supervised learning method that is a member of the classification and regression tree family. Based on the information at hand, it constructs a tree-like model of decisions and their outcomes. To maximize the homogeneity of the generated subsets, the method iteratively divides the dataset into smaller subsets based on the most important attribute. A tree-like structure is produced because of this process, which can be used to anticipate the outcomes of new data (Bansal, Goyal and Choudhary, 2022). [**APENDIX B – Decision Tree Structure**](#_APENDIX_B_–) |
| K nearest neighbor algorithm | The KNN algorithm is a supervised learning method that generates predictions based on how similar new data instances are to training instances. The approach employs the distance between the points to calculate the similarity between all the instances that are stored as points in a multi-dimensional space. When a new instance is introduced, the algorithm locates the k-nearest neighbors in the training set and places the new instance in the class that is most prevalent among its k-nearest neighbors. When a new instance is introduced, the algorithm locates the k-nearest neighbors in the training set and places the new instance in the class that is most prevalent among its k-nearest neighbors. Several fields, including recommender systems, have successfully used the KNN algorithm (Sagdic et al., 2020). |
| Support Vector Machines | The SVM is a supervised learning method that seeks to locate a hyperplane that maximizes the margin between the classes. Using kernel functions, the SVM can handle both linear and nonlinear classification issues. The SVM algorithm is especially well suited for datasets with many characteristics and few data examples. If the regularization value is not properly adjusted, the SVM algorithm may be prone to overfitting (Gil, Freudenthaler and Natschläger, 2018). |

Table 4 Machine Learning Algorithm

## 2.11 Evaluation Techniques

|  |  |
| --- | --- |
| Metrics | Description |
| Accuracy | A measure of accuracy is the percentage of correctly classified data instances out of all instances. For classification algorithms like the Random Forest Classifier, it is frequently employed as an evaluation metric. High accuracy means that the classifier can accurately categorize a significant fraction of the data instances, whereas low accuracy means that many examples are misclassified. |
| Precision | A measurement of precision is the percentage of suggested things that are relevant to the user's interests. For content-based filtering algorithms, it is frequently employed as a measurement parameter. While low precision suggests that the recommendations are not relevant, high precision suggests that the user will find the suggested items to be highly relevant. Precision is determined by dividing the number of true positives—that is, relevant items that are correctly recommended—by the total number of items recommended. |
| Recall | Recall is a measurement of the percentage of pertinent items that are suggested to the user. It is frequently employed as a statistic for content-based filtering algorithms. Low recall means that many relevant items are missed, but high recall shows that the algorithm can recommend a significant amount of the user's relevant items. Recall is calculated as the proportion of true positives to all relevant items. |
| F1 | A recommendation system is evaluated fairly using the F1-score, a metric that combines precision and recall. Precision and recall are determined as harmonic means, which gives each statistic equal weight. While a low F1-score suggests that the algorithm is deficient in either precision or recall, a high F1-score indicates that the algorithm possesses both. |
| ROC AUC Score | A statistic used to assess the effectiveness of binary classification algorithms is the ROC AUC score. By displaying the True Positive Rate (TPR) versus the False Positive Rate (FPR) at various classification thresholds, it assesses the model's capacity to distinguish between positive and negative classes.  The classification threshold, which determines whether a predicted probability belongs to the positive or negative class, is changed to produce the ROC curve. The proportion of positive events that are correctly categorized as positive is known as the TPR, whereas the proportion of negative events that are wrongly labeled as positive is known as the FPR. Higher numbers denote greater performance. The AUC score is the area under the ROC curve and ranges from 0 to 1. |

Table 5 Evaluation Techniques

## 2.12 Benchmarking

The benchmarking of a system could be done by comparing the performance of the model with other different models in accordance with the requirement of the system. In fact, the dataset used during the process of training to do a better recommender system by selection of one algorithm or based on the hybrid approach could be also a part of benchmarking, Since the dataset will be specific for requirement that is based on the IT industry for Sri Lanka. Moreover, solving salary discrimination has not been done by any of the researcher with the aid of recommendation system accordance to the author study in the domain background, as the result the uniqueness of the system could also be one of the benchmarking.

## 2.13 Chapter Summary

In this chapter the author has spoken about the problems in the research domain, such as in the Financial as well as the recommendation system. In addition, the author has also explained about the similar products that are already been researched in classification, recommendation techniques as well as prediction. The chapter also clearly explains the possible types of recommendation techniques and the machine learning algorithm which could be used in the recommender system along with the how and how well the build models could be evaluated.

# Chapter 03: Methodology

## 3.1 Chapter Overview

In this chapter the author will be discussing the research methodology, design methodology of the proposed system and evaluation methdodgy. Moreover, the project deliverables with the relevant dates, under the risk management the risk and mitigation plans will also be discussed by the author.

## 3.2 Research Methodology

|  |  |
| --- | --- |
| Philosophy | Research philosophy can be categorized into different types which are Positivism, Interpretivism and Pragmatism. Based on this research basis and by analysis the data collection and data analysis the author has chosen Positivism since this thesis has both quantitative and qualitative information’s. |
| Approach | As per the research approaches there are 2 types. The first is Deductive and the next is inductive. The author, in this research is implemented a deductive based approach, since the recommendation systems have already existed, since to prove that recommendation system will be helpful for the salary domain as well and will help the individuals. |
| Strategy | This research is full of quantitative and qualitative information; hence the strategic approach is to have questioners from the individual of each company based on the salary discriminations. Moreover, interviews will also be conducted on how well the system will be helpful for individuals; individual going to start a new job, or an individual is going to switch to higher position in his job career. |
| Choice | The choice of methodology is decided based on the quantitative and qualitative of the research. The proposed research will be using the multi-Method since both quantitative and qualitative has its equal measure in this thesis. |
| Time Horizon | Longitudinal has been selected as the time Horizon for this research since the data gathered and tested for a long period of time. |
| Technique and Procedures | To collect Data and different kind of analysis techniques, observations, reports, surveys, questionnaires, documentation will be used. |

Table 6 Research Methodology

## 3.3 Development Methodology

There are different types of development methodologies that an author can select on. For this research and implementation, the methodology that has been chosen by the author is the **Spiral Methodology.**

This type of methodology includes the combination of Agile and Waterfall methodology, since there should be rapid early risk detection on all process of the stages in the development life cycle goes on.

### 3.3.1 Design Methodology

The design methodology that the author has chosen is Objected Oriented Design, which is the most used design pattern in any software development project. And helps to reuse the objects or many codes in this type of approach.

### 3.3.2 Evaluation Methodology

The system will be tested and evaluated using different metrics such as Mean Squared Error, Mean Absolute Error, Precision, Recall, False positive rate, these benchmarking is also done in different recommendation system (Shani and Gunawardana, 2011). In addition to it this system will help to overcome the main problems of recommendation system that is Cold start, data sparsity, accuracy and help to overcome the salary discrimination problem.

## 3.4 Project Management Methodology

The author has selected the agile Prince2 methodology based on this thesis. It is due to since the project focuses on both management and delivery at the same time, hence this will allow the author to complete the relevant expected task and document on time. In fact, this approach could change tools and technology if the requirement constantly keeps changing, which could be an added advantage of selecting Agile Prince 2.

### 3.4.1 Gantt Chart

Please refere to [APPENDIX C -Gantt Chart](#_APPENDIX_C_-)

### 3.4.2 Deliverables

|  |  |
| --- | --- |
| Deliverables | Date |
| Project Proposal Documentation | 4th November 2022 |
| Literature review Documentation | 16th March 2023 |
| Software Requirement Elicitation | 25th January 2023 |
| Prototype Implementation | 6th February 2023 |
| Thesis | 25th February 2023 |
| Final Research Paper | 25th May 2023 |
| Final Viva | 20th May 2023 |

Table 7 Deliverables

### 3.4.3 Risks and Mitigation

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Severity** | **Frequency** | **Mitigation Plan** |
| Loosing access to the documentations or codes that has been implemented | 5 | 4 | Keep the code backed-up in version control and use google drive to back up the documentations. |
| Unable to complete the topics as per the plan | 4 | 3 | Work on the topic as per the priority basis. |
| Unable to complete the research implementation due to illness. | 5 | 5 | Have proof about the sickness. And if it goes worse then apply a self-mitigation. |
| In-depth knowledge in the domain of Salary, need to be discussed as per the recommendation are the salary up to date. | 5 | 3 | Discuss with the domain expert and discuss with few companies and validate the salary is it up to date. |

Table 8 Risks and Mitigation

## 3.5 Chapter Summary

This chapter illustrates the research methodology according to philosophy, approach, strategy, choice, time horizon and finally technique and procedures sub-topics were discussed. In the development methodology the author discussed the evaluation, design methodology of the proposed system and finally about the project deliverables and possible risk that the project could face during the implementation period.

# Chapter 04: Software Requirement Specification

## 4.1 Chapter Overview

This chapter focuses on finding potential project stakeholders by examining all potential points of system contact using a rich picture diagram, gathering their perspectives to analyze and develop potential expected use cases, functional and non-functional prototype requirements.

## 4.2 Rich Picture

A diagram of a system

Description automatically generated with low confidenceThe system's detailed rich picture diagram shows the identified stakeholders as well as how the system interacts with the environment, which was not known in the beginning. A graphical representation of all the stakeholders and how the systems relate to them may be found in the picture below.

Figure 3 RICH PICTURE Diagram

## 4.3 Stakeholder Analysis

In Stakeholder Viewpoints, the Stakeholder Onion Model depicts well-known stakeholders connected to the proposed system along with a description of each stakeholder's involvement in the system.

### A diagram of a diagram Description automatically generated with low confidence4.3.1 Onion Model

Figure 4 Stakeholder Onion Model

### 4.3.2 Stakeholder viewpoints

|  |  |  |
| --- | --- | --- |
| **Stakeholder** | **Role** | **Benefits/Role Description** |
| Product  Developer | Financial  Beneficiary | Develops the system. |
| Product Owner | The owner of the building system. |
| Hackers | May manipulate with the Data that is stored. |
| Competitors | Negative  Stakeholders | May build competing products better than the current system. |
| Supervisor | Advisory | Giving guidance on how well to implement the system and improvements that need to be made. |
| Labor Union | Social  Beneficiary | Will be able to ensure that fair wages are being met based on the recommended data. |
| Government | It will help to ensure that the salary recommendation system complies with the labor laws and regulations. |
| IT Professionals  **(**  Software Engineers,  Technical Leads  Software Architectures  Quality Assurance  Devops Engineer  Data scientists  UI/UX Engineer  **)** | Operational  Beneficiary | Will be able to gain information about the salary that an individual will be able to get based on individual performance. |
| IT  Undergraduates | Will be helpful to know on how much salary will an undergraduates gain as an individual will be joining the IT industry |
| IT Postgraduates |  | This system will be helpful to bargain or get an idea on how much an individual will gain according to their study and work experience. |
| Human Resource | Functional  Beneficiary | This will help HR to forecast the expenses which might be incurred during the period. |
| Project Manager | Administrator | This will enable the manager to forecast the expenses of the employee’s salary to analyze the profits that the Manager could gain in the future. |

Table 9 Stakeholder viewpoints

## 4.4 Selection of Requirement Elicitation Techniques

In this section the author will give a brief about the methodologies that have been used to gather information on the development of this research project. There are different methods to acquire requirements of the project, but as per the author of this project the selection of requirement elicitation methodologies is questionnaire, Brainstorming and interview. The reasons for selecting the specific requirement elicitation methodologies are mentioned below.

|  |
| --- |
| Method 01: **Questionnaire** |
| The questionnaire has been used by the author to gather information from a wide group of stakeholders from the proposed system. In addition, by using this methodology the author will be able to analyze if the proposed solution will be helpful for the target audience. |
| Method 02: **Brainstorming** |
| Brainstorming was used by the author to come up with ideas on how to improve the system and make it user friendly since the target audience should be able to make use of this proposed system. With the aid of using this technique the author will be able to break down the complex problem into smaller ones, which in turn implement them step by step. The brainstorming discussion was done with the supervisor, colleagues, and a few of the IT professionals. |
| Method 03: **Interview** |
| The interview was conducted by the author to gather a little information about the domain being worked on in the proposed system. By means of interview possible requirements could be identified and how to be solved by using machine learning could be analyzed. This will make sure to understand the drawbacks and challenges that will be faced while implementing the proposed system. |
| Method 04: **Prototyping** |
| Before moving on to the level of implementation, prototyping was used. This will be used mostly to improve system usability. It aids in developing a complete understanding of the system and its features. Author can gain a thorough understanding of what should be done |
| while prototyping and what shouldn't be added to the system. This results in a low error rate and minimal misunderstanding during implementation. |

## 4.5 Discussion of Results

### 4.5.1 Questionnaire

|  |  |
| --- | --- |
| Question | Are you from an IT industry? |
| Aim of question | The scope of the proposed system is to recommend salary only to the IT industry; hence this question proves that an individual is from the IT industry. |
| Observations | |
|  | |
| Conclusion | |
| This question proves the target audience that has filled in the questionnaire, since the system is being implemented only in the IT industry sector. As per the result 98% of the questionnaire is being filled by the IT industry individuals. That proves it has accomplished the aim of the questionnaire | |
| Question | Have you ever thought that the amount of remuneration/Salary that you're earning for the level of experience that you have, is less compared to other people of the same level of experience? |
| Aim of question | This to understand that will the experience level of an individual affect the amount of remuneration earned for a period. |
| Observations | |
|  | |
| Conclusion | |
| The question proves that there are possibilities for employees that of the same level of experience could gain a lesser amount in the IT industry, which could be in the same company or a different company. As per the survey 80% of individuals have agreed with this statement. | |
| Question | Do you think there is still salary discrimination in the corporate world? |
| Aim of question | To analyze if the employee of the company will be given a different salary amount even if an individual has the same level of experience but they are of different gender |
| Observations | |
|  | |
| Conclusion | |
| The pie chart above shows that 92% agree with the fact that there is salary discrimination in the IT industry. It makes sure that one of the research gaps in the proposed system can be fulfilled by the author, by recommending the best salary for the individual which will help the user to bargain based on this data. | |
| Question | Will a salary recommendation system be helpful for you to decide your next remuneration? |
| Aim of question | To make sure the system which is going to be built by the author is helpful for the IT industry individuals. |
| Observations | |
|  | |
| Conclusion | |
| Based on the survey it reveals that 60% of individuals agree with having a salary recommendation system. 24% of the individuals strongly agree about this system, which will help them to make some kind of decision in their career by the system. 8% of Neutral and 8% disagree with the system, but it clearly proves the individuals will have a positive impact by this system. | |
| Question | Do you wish to share your salary? |
| Aim of question | To identify if the individuals are willing to share their salary amount that is being earned during their working period. |
| Observations | |
|  | |
| Conclusion | |
| According to the information of the questionnaire it indicates that 56%, that is almost half of the individual, is willing to share their salary details, which is also an advantage for the author that will help to get the relevant salary details for implementation purposes. | |
| Question | What position are you working in the IT industry? If not working currently, please specify your old designation |
| Aim of question | To get the job role that an individual is currently in for the people who has agreed to share their salary details. |
| Observations | |
|  | |
| Conclusion | |
| As per the analysis 46.4% of the individuals are from the internship role and 10.7% are from Associate Software Engineers and few others from different IT roles. This could be used to make sure the salary amount that individuals will gain according to the role they work on. | |
| Question | What is the current level of experience do you have (In Years)? |
| Aim of question | To identify the year of experience the individual has been working to get the position that an individual is currently working on, and the amount of salary that has been earned for the level of experience the individual has |
| Observations | |
|  | |
| Conclusion | |
| The result of this question depicts 60.7% are in the 1-year experience stage and 39.3% are in the 1-3 year of experience. This data could be used by the author to make an effective way of the recommendation model that will be built to cross check the reliability of the system. | |

Table 10 Questionnaire

### 4.5.2 Brainstorming

|  |  |
| --- | --- |
| Criteria | Findings |
| Implementation of the Recommendation System | To build the system, a proper dataset is needed which must be preprocessed and well-maintained data with proper analysis of each column and how it will help to recommend the data. To make the system recommendation algorithm try out different kinds of algorithms and choose the best that gives out the highest accuracy and proper evaluation to the system. |
| Evaluation of the proposed system | As per doing this technique it has helped the author to identify that the proposed system will be very much helpful for the target audience since it will be able to overcome salary discrimination and low pay remuneration for the same level of experience, as a result will help the target audience using the system to bargain and make a better decision in terms of salary. |
| Evaluation of the UI design | Since the focus of the system is to recommend salary, the UI should be very much user friendly since it allows users to enter the salary as well as get the salary based on following user input the system allows to enter. |
| Feasibility | Up to a certain extend this system will be feasible to do, but also it could have few drawbacks since there are whole lot of factors that will be affecting the salary; hence it could only be a kick start to know for the user that this is salary the individual will be gaining, but it could be changed over the future period. |

Table 11 Brainstorming

### 4.5.3 Interview

|  |  |
| --- | --- |
| Theme | Analysis |
| Scope of the Proposed System | As per the discussion, the interviewee thought that the proposed system will be very much helpful to the target audience since it will overcome few of the problems in the IT industry as per the specification stated. |
| Data analysis | As per the Domain expert the data should be well preprocessed to solve and get the best output for the system. By doing appropriate preprocessing and analysis the data properly will help to make sure that the model created could be of very high accuracy. |
| Techniques used to build Recommendation system | According to the ML expert the recommendation system doesn't always have to have a model or algorithm; it could be done by the  help of content-based or collaborative based which is mostly made by the help of cosine similarity but not being used of algorithm such as KNN or SVD. Hence the technique that should be used will be determined by the dataset, developer is using and according to the requirements of the system. |
| Analysis of the Salary recommendation system | According to the discussion, salary recommendation is helpful for the target audience but won’t be able to always rely on the proposed system, since the salary will be changing overtime, even though the proposed system is feasible, as per the discussion salary recommendation should be considered how to update the value over the period. |

Table 12 Interview

## 4.6 Summary of Findings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Findings | Questionnaire | Brainstorming | Interview | Prototype |
| Will the proposed system be very much helpful for the target audience | ✓ | ✓ | ✓ | ✓ |
| Will the system resolve salary discrimination | ✓ |  | ✓ |  |
| Can the proposed system make better decision for bargaining your salary |  | ✓ | ✓ |  |
| What type of recommendation system should be done for the proposed requirements |  | ✓ |  | ✓ |
| Proper tools should be selected to implement the system |  | ✓ | ✓ | ✓ |

Table 13 Summary of Findings

## 4.7 Context Diagram

A context diagram is a graphic depiction of a system and its external components that demonstrates their communication and interaction.

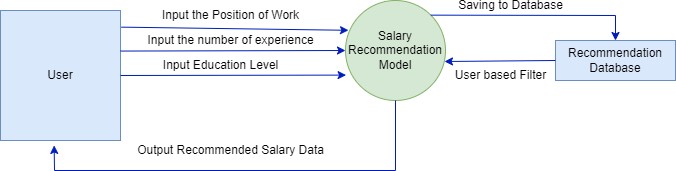


Figure 5 Context Diagram

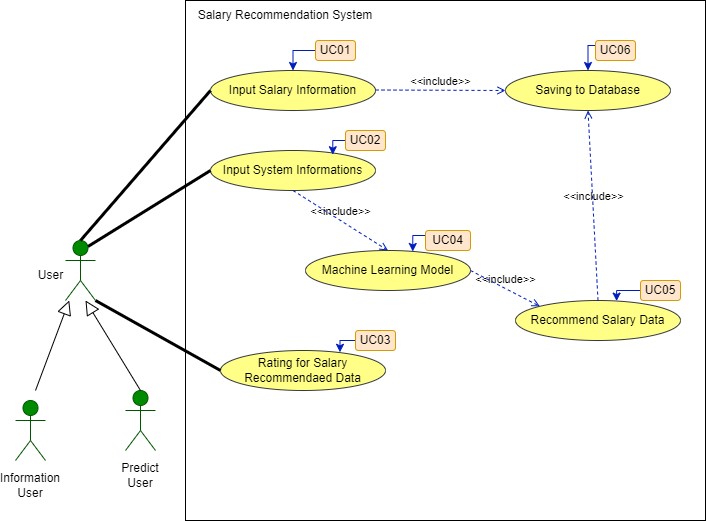
## 4.8 Use Case Diagram

A use case diagram demonstrates how actors and use cases relate to one another in a system. As per the Use case diagram depicted below it indicates the number of actors and the roles that the Users will be playing in the system.

There are 2 different kinds of Actors according to the system.

1. Information User – This user will be able to enter Salary details in the system, so that it will help to improve the system's recommendation, with the updated salary which the user enters.
2. Predict User – These User will use the system, to get Salary recommendations for their relevant inputs the User has given into the system to predict the result.

Figure 6 Use case diagram.



## 4.9 Use case Description

### 4.9.1 Use case of Input Salary Information.

|  |  |  |
| --- | --- | --- |
| Use Case | Input Salary Information | |
| Use case Id | UC - 01 | |
| Description | Information User entering Salary Details | |
| Primary Actor | Information User | |
| Supporting Actor | - | |
| Pre-Conditions | - | |
| Trigger | User wants to enter the Salary Details | |
| Main flow | Actor | System |
| 1. User opens the web application. 2. User opens the Salary Information Tab 3. Users enter the inputs that are needed. 4. Success Message will be displayed to the User | 1. The system will request for the user to enter the Salary information. 2. 5. System processes the Data entered |
| Exception flow | Actor | System |
| 1. User tries to submit the data without entering important information | 1. System will display error message asking to enter important information |
| Post conditions | System will process the data entered, it will be stored in the Database and if the information is successfully stored, A success message will be displayed to the User. | |

Table 14 Use case of Input Salary Information

### 4.9.2 Use case for Recommend Salary Data.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Recommend Salary Data | | |
| Use case Id | UC - 05 | | |
| Description | The Salary details will be suggested to the user based on the input that is pooled into the system. | | |
| Primary Actor | Predict User | | |
| Supporting Actor | - | | |
| Pre-Conditions | The necessary inputs should be entered. The recommendation model should be loaded | | |
| Trigger | User wants to get the information about the salary specific for the  information given. | | |
| Main flow | Actor | | System |
| 1. User opens the web application. 2. Users open the salary Recommendation Tab 3. Users enter the input data needed for the recommendation. 4. Users will be suggested with the salary information with the relevant information provided. | | 1. System will request for the user to enter the relevant information. 2. System will process the data entered to recommend salary details. |
| Exception flow | Actor | System | |
| 1. User tries to submit the data without entering important information | 1. System will display error message asking to enter important information | |
| Post conditions | System will process the data into the recommendation model, and suggest relevant salary information for the user, according to the inputs the user has given. | | |

Table 15 Use case for Recommend Salary Data

### 4.9.3 Use case for Rating for Recommended data

|  |  |  |
| --- | --- | --- |
| Use Case | Rating for Recommended data | |
| Use case Id | UC - 03 | |
| Description | The Salary that is being recommended for the User will be able to be given rating by the User | |
| Primary Actor | Predict User | |
| Supporting Actor | - | |
| Pre-Conditions | The necessary inputs should be entered. The recommendation model should be loaded | |
| Trigger | User wants to get the information about the salary specific for the  information given. And then the Rating could be provided by the User | |
| Main flow | Actors | System |
| 1. User opens the web application. 2. User opens the Salary Recommendation Tab 3. Users enter the input data needed for the recommendation. 4. Users will be suggested with the salary information with the relevant information provided. 5. Users will be able to give ratings for the suggested Information, to make sure of the correctness of the   recommendation. | 1. System will request for the user to enter the relevant information. 2. System will process the data entered to recommend salary details. |
| Exception flow | Actor | System |
| 1. User tries to submit the data without entering important information | 1. User tries to submit the data without entering important information |
| Post conditions | System will process the data into the recommendation model, and suggest relevant salary information for the user, according to the inputs the user has given. At the end the User can also give the ratings to make sure the correctness of the information. | |

Table 16 Use case for Rating for Recommended data.

4.10 Requirements

The MoSCoW technique has been used by the author to determine the priority level of the system based on their importance.

|  |  |
| --- | --- |
| Priority Level | Description |
| Must(M) | This represents the Core functionality of the system; the functionality must be implemented. |
| Should (S) | The functionality is good to be implemented which add value to the system, but it is not of a high priority. |
| Could (C) | These requirements are optional which are not essential to the project scope. |
| Wont (W) | The functionality which will not be considered as a high priority of the project scope level. |

Table 17 Requirement Description

### 4.10.1 Functional Requirement

|  |  |  |  |
| --- | --- | --- | --- |
| FR ID | Requirement | Priority Case | Use case |
| FR01 | Users can enter salary information details. | S | UC01 |
| FR02 | User inputs data to get the salary recommendation data based on the parameters that the user has pooled into the system. | M | UC02 |
| FR03 | Saving the recommended data based on the user input into the database | M | UC06 |
| FR04 | Saving the salary information provided by the information user to the database | S | UC06 |
| FR05 | Recommending relevant salary data for the user’s entered inputs. | M | UC05 |
| FR06 | Reinforcement learning could be used to improve the recommendation model | C | NA |
| FR07 | The system allows the user to give ratings to the recommended salary data. | S | UC03 |
| FR08 | The salary prediction of the future dates. | W | NA |
| FR09 | The system shows error messages for inappropriate use of the system | M | NA |
| FR10 | Request for feedback for the recommendation suggestions from the user. | C | NA |
| FR11 | Adjusting the relevant salary recommended salary to the latest inflation rate. | S | NA |

Table 18 Functional Requirement

### 4.10.2 Non-Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| NFRID | Requirement | Description | Priority Level |
| NFR01 | Performance | The performance of the System is very important, Since the user should be able to get the recommendation data as quickly as possible which will not be wasting time being idle, to get the recommendation data. | Important |
| NFR02 | Usability | The user will be suggested with the relevant salary data based on the parameters the user has pooled into the system, therefore the user must be able to use the system in a friendly and easy way, which will help the user to make important decisions according to the data recommended. | Important |
| NFR03 | Security | The application must prevent any attackers from manipulating the salary information and user inputs. This could be overcome by means of testing. | Desirable |
| NFR04 | Maintainability | The documentation of the implementation and other information should be properly entitled so that the out scope of the project can be implemented in the future, by refereeing the information given. | Desirable |
| NFR05 | Reliability | The suggestion of the system should be realistic so that the user will be able to get some knowledge of the data recommended. |  |

Table 19 Non-Functional Requirement

## 4.11 Chapter Summary

To better comprehend the system's stakeholders, a Rich Picture Diagram was created in this chapter to show how the system interacts with society. The stakeholders and the flow of influence from each stakeholder were represented using the onion model. Techniques for requirement gathering were used to collect all the necessary information and the views of potential system stakeholders. Finally, using the knowledge gained from the requirement elicitation approaches, the use cases, functional requirements, and non-functional requirements for the system were established.

# Chapter 05: Social, Legal, Ethical and Professional issues

## 5.1 Chapter Overview

In this chapter it covers the social, legal, ethical, and professional issues that could take place during the implementation of the proposed system.

## 5.2 SLEP issues and Mitigation

|  |  |
| --- | --- |
| Social | Legal |
| * Whilst collecting information through questionnaires the user’s information was not obtained and only the summary of the whole response was recorded in the thesis. * Interviewers who were interviewed requested to keep their details anonymous, hence the details of the interviewers were not revealed in the thesis. | * The tools and framework that was used to implement the proposed system are used under an open-source license. * The survey that was send across the IT professionals and Undergraduate Student of IT field did not gather any personal information. * The survey that was send across the IT professionals and Undergraduate Student of IT field did not gather any personal information. |
| Ethical | Professional |
| * The data or information that was researched was properly cited and given credits to the author and to avoid plagiarism. * Individuals who completed the survey for the proposed system, was well informed about the project and the contribution that individual would give. | * A proper and well standard was followed during the process of the research. * The software and tools that were used by the author was not illegal or pirated, either it was open-sourced or student licensed software. * The implementation and the proposed system were well documented by following the best practiced in both implementation as well as during the documentation. |

Table 20 SLEP issues and Mitigation

5.3 Chapter Summary

In this chapter the author has clearly stated the possible social, legal, ethical, and professional issues which are related to this research project and how has these identified issues were solved.

# Chapter 06: System Architecture and Design

## 6.1 Chapter Overview

The design choices made in this chapter, based on the requirements acquired, resulted in an architecture that is suitable for implementation. When describing the justification for design decisions, high-level design, low-level design, design diagrams, and UI wireframes have been used to demonstrate how the design goals are intended to be accomplished.

6.2 Design Goals

|  |  |
| --- | --- |
| Design Goals | Description |
| Performance | An application's performance must be considered when designing a system. The system's performance is measured by how quickly it completes a task without lagging. As a result, when creating the system, this design goal is also considered. |
| Reusability | The components that are involved to build the system, should easily be extended for other developers, trying to modify or improve the system. |
| Adaptability | In any system, there would be future enhancement or removing unwanted features, therefore the components that are involved to build the system should easily be updated such as changing of the algorithm or APIs etc. as a result it should also not break the core functionality of the system. |
| Scalability | In the production environment the system should be available for all the users that try to access concurrently; hence the backend server should be able to handle all the possible requests. |
| Correctness | The output of the system should be the best fit regarding quality and correctness of the data which is recommended to the user according to the availability of data. Giving the best output will help the user to use for their future enhancement in their career. |

Table 21 Design Goals

6.3 System Architecture Design

### 6.3.1 Three tier Architecture diagram

The system is designed by Three-tier software architecture. Three-tier architecture has 3 main tiers, the presentation tier, the Logic tier, and the data tier.

### 6.3.2 Discussion of tiers

Figure 7 Architecture Diagram

According to the above diagram which is represented in the approach for the ease of understanding. The author has chosen the implementation of the proposed system to be built in a microservice architecture. The reason for implementing the proposed system in this manner is because it allows to improve maintainability, which means the developer will be able to find bugs or any point of failures easily, as a result the fixes could be done separately.

6.3.2.1 Presentation Tier

1. Salary information UI (User Interface) – This UI displays the user with necessary data inputs which will allow them to save the information. The data inputs that are displayed to the user are depicted as Input Salary information in the above diagram.
2. Recommend Salary Data UI – This UI will allow the user to input a few specific parameters as a result, this will recommend the user with relevant information that has to be shown according to the model that is trained. The information input that the user must select is depicted as User input data for recommendation in the above architecture diagram.
3. Display result UI – This UI will display the best prioritized information in accordance with the model for the captured data which is entered by the user.

6.3.2.2 Logic Tier

* + - * 1. Salary details save API – This is one of the backend API which will be request once the user is trying to save the details of the salary information, during this scenario few validations will be taking for the entered details so that the collected information will be able to be used for future recommendations therefore improving the recommendation and improving the user decisions.
        2. User recommendation API – This is another backend API, requested when the user is trying to get the recommendation result for the relevant inputs that has been pooled into the API.
        3. Recommendation Engine – This is the model and the content-based technique that is built by the developer to get the relevant information according to the user input.
        4. Recommendation Model – This is the model which is done by the author to get salary information based on the user’s inputs.

6.3.2.3 Data Tier

The database tier will save the information about the new salary that will be entered by the user, which will help the developer to make better content-based recommendations for the end user. Moreover, the database layer will also save the recommended data hence for future enhancement of the system, it will help to implement another technique in the recommendation system which is user-based filtering to improve the performance of the system.

## 6.4 System Design

### 6.4.1 Choice of Design Paradigm

There are 2 possible kinds of design paradigm used to implement a system, those are SSADM (Structured System analysis and design method) and OOAD (Object oriented analysis and design method). SSADM and OOAD have their own advantages and disadvantages, but when building up a system, the developer will have to properly analyze and decide on which time of design approach the developer is going to focus on. As per this proposed system, the author/developer has decided to choose OOAD. OOAD implementation is very easy, it uses objects and classes to model the entities and their relationship, which gives a detailed understanding of the objects. More importantly the components of the software will be able to be reused, which is an added advantage.

### 6.4.2 Component Diagram

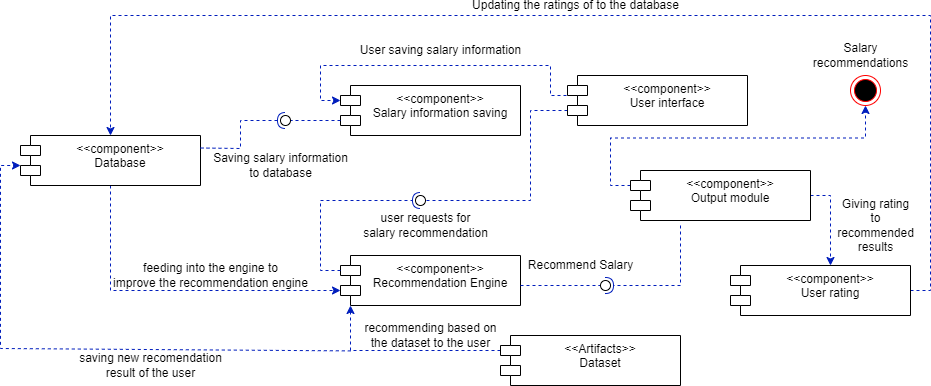
The following diagram depicted below describes the components of the proposed system, and their relations on how it will communicate between each of the processes and the way users interact with the system.

Figure 8 Component Diagram

### 6.4.3 Class Diagram

Since the author has selected OOAD the class diagram of the proposed system represented below describes the classes involved in implementing the proposed system.

There is an inheritance according to the user, the user can give information to save a salary or else a user can get recommendation based on its input, since save\_information\_user and recommendation\_user have common attributes, it has been inherited by the User class.

On the other hand, the rating class depends on the user. The user will give a rating for the relevant recommendation show in the UI, which will update the salary information accordingly.

### 6.4.4 Sequence Diagram

Figure 9 Class Diagram

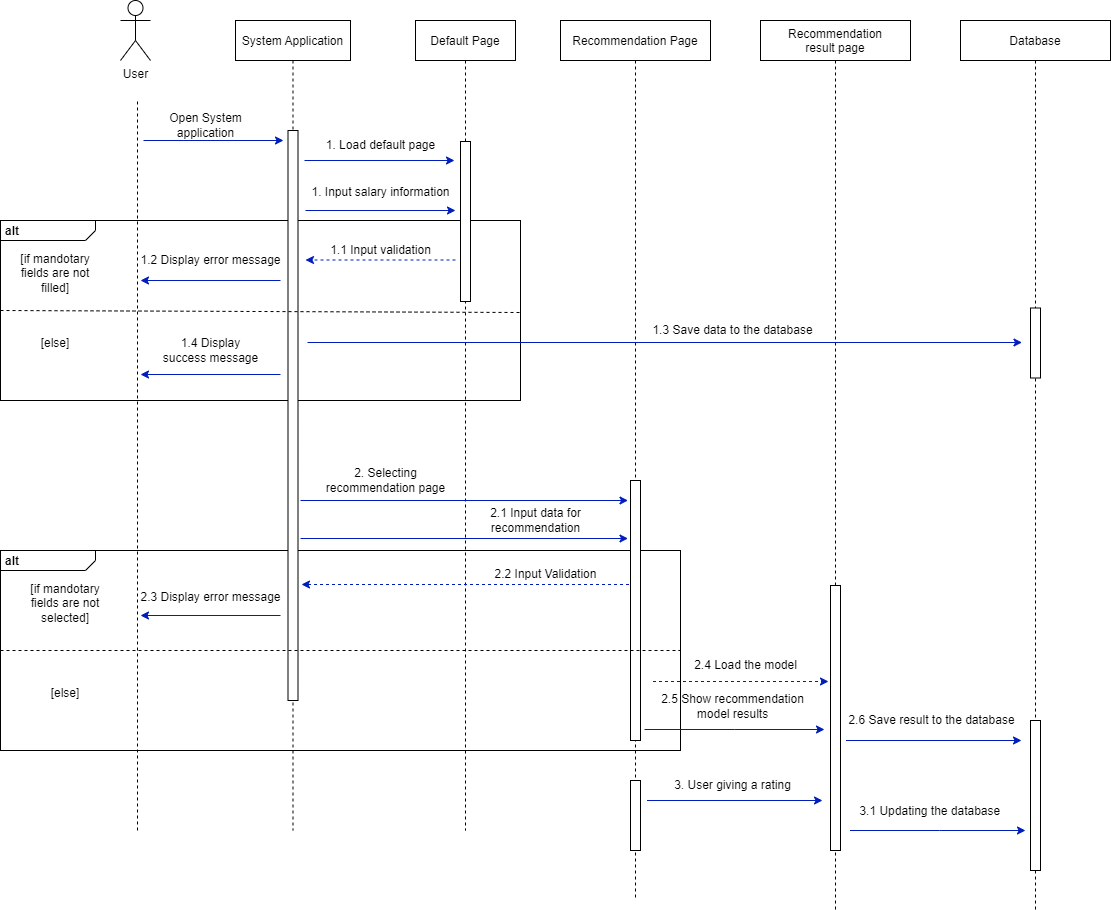
The below diagram illustrates the sequence diagram of the proposed system, the sequence of which the user will be interacting with the system application implemented by the author.

Figure 10 Sequence Diagram

### 6.4.5 UI DesignA picture containing text, screenshot, software, number Description automatically generated

Figure 11 Recommendation UI

A picture containing text, screenshot, software, number

Description automatically generatedA close-up of a document

Description automatically generated with low confidence

Figure 12 Salary Input UI

Figure 13 Salary Output card

### 6.4.6 Process Flow Chart

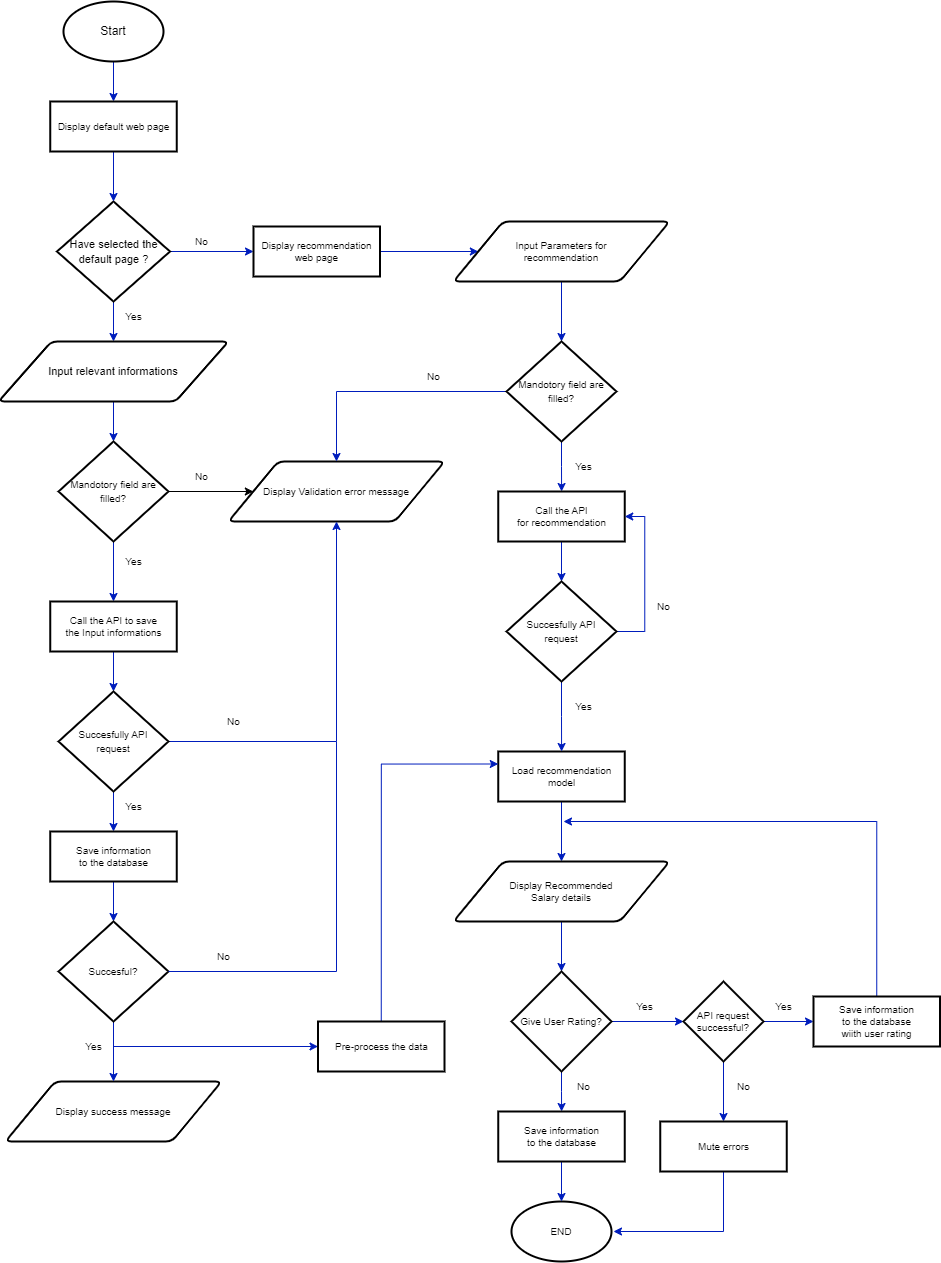
The visual representation of the flowchart being depicted below, illustrating the algorithm flow and decision-making structures which will be implemented for the proposed system by the author.

Figure 14 Process flow chart

## 6.5 Chapter Summary

The system architecture and design chapter described the system design procedures that were followed while taking the findings from earlier chapters into account. The design objectives, the three architectures that were chosen, as well as the class diagram, sequence diagram, and wireframes of this system, have all been covered in this chapter by the author. The UML design patterns were used for all the designs. And the implementation of the system's prototype will be covered in the following chapter.

# Chapter 07: Implementation

# 7.1 Chapter Overview

This chapter discusses implementation, which involves choosing the appropriate technology. Under the heading "Technology Selection," there is an illustration of the chosen frameworks, languages, IDEs, and libraries. The author will then talk about how to choose the data that will be used to train and test the ML algorithm. Following that, the rationale for library, IDE, and framework choices will be covered. The implementation of fundamental features will then be covered together with the pertinent code samples.

## 7.2 Technology Selection

### 7.2.1 Technology stack

A screenshot of a computer

Description automatically generated with low confidenceThe technologies that have been used by the author to implement the proposed system are depicted below.

Figure 15 Technology Stack

Window operating will be the default system that will be used by the author, for implementation, documentation, and research purposes.

### 7.2.2 Data Selection

The proposed system by the author is a data science project, therefore the dataset is the core component to build the model for analysis and produce the best possible salary recommendation for the users using the system.

The dataset used by the author to develop the recommendation model is obtained from tech salary.lk, a website that is used to collect salary of employees working in the IT industry. The other form of collecting the dataset was by using the survey, which allowed the IT industry colleagues to enter the salary details if individuals were willing to share their salary information. Moreover, a few data were also collected from the glass Door with the help of web scraping.

The dataset, collected in different ways, has been combined by the author to develop the proposed system and recommend salary details according to the user’s inputs that are passed to the model.

### 7.2.3 Selection of Development Framework

|  |  |
| --- | --- |
| Framework | Justification |
| React JS | React JS is a popular frontend JavaScript library which is used to build user interfaces. The reason for choosing this framework is due to component-based architecture, which allows the application to break down into smaller components that makes it easier to maintain and manage for the developer.  React JS has a larger community hence it’s an added advantage since the developer could seek help when building the proposed system. |
| Flask | It is very much easy to build APIs for python and connect machine learning models. |

Table 22 Selection of Development Framework

### 7.2.4 Programming Language

Python is the programming language which was used to do the machine learning model. Python has plenty of libraries. These libraries can be used to build the recommendation model easily for the developer, also use the existing algorithms to be tested and compare them, to analyze which is the best algorithm for this proposed system.

JavaScript was the programming language used to build the frontend, to show dynamic content and allow high interactable and better user experience.

### 7.2.5 Libraries

|  |  |
| --- | --- |
| Library | Justification |
| Matplotlib and seaborn | These libraries are used for data visualization and analysis. |
| Pandas | Pandas’ data frames enable a wide range of data analysis features, including cleaning, transforming, filtering, sorting, and manipulating data. |
| Scitkit-Learn | Used to create similarity matrices between items and vectorize text for recommendations. |
| React | A user interface library that makes it simple to create interactive websites. Since the front end will be where consumers interact with the system, it was crucial to create a user-friendly interface. Thanks to Reacts huge number of functionalities, this was simple to accomplish. |
| NumPy | This library will be helpful to do mathematical functions for multi- dimensional arrays and matrices. |

Table 23 Libraries

### 7.2.6 IDE

|  |  |
| --- | --- |
| IDE | Justification |
| VSCode | A powerful tool for front end development since the IDE could be updated with different kinds of extensions. Also, it is simple and easy to use for all the developers. |
| Jupyter | Used for model training and data science model testing purposes. |
| PyCharm | This IDE is easy to use for the flask development framework, it has a user- friendly interface with advanced features which helps to write high quality code faster with few errors. |

Table 24 IDE

### 7.2.7 Summary of Technology Selection

|  |  |
| --- | --- |
| Component | Tools |
| Programming Language | Python, JavaScript |
| Development Tools | Flask |
| UI Framework | React |
| Libraries | NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn |
| IDE’s | Jupyter notebook, VsCode, PyCharm |
| Version Control | Git, GitHub |

Table 25 Summary of Technology selection

## 7.3 Implementation of Core Functionalities

The below code snippets represent the core functionalities of the salary recommendation system.

### 7.3.1 Data preprocessing

### A screenshot of a computer code Description automatically generated with low confidence7.3.1.1 Removing unwanted Columns.

Figure 16 Removal of unwanted Columns

The above snippet represents the removal of the unwanted column in the dataset, which is not needed to train the classifier model.

### A picture containing text, screenshot, font Description automatically generated7.3.1.2 Company size preprocess

Figure 17 Company size preprocess

The diagram above shows the preprocessing of the company to make into a common label, which will be easy to identify or make the model class easily.

### A screenshot of a computer Description automatically generated with low confidence7.3.1.3 Conversion to common currency ‘LKR’

Figure 18 Conversion to common currency LKR

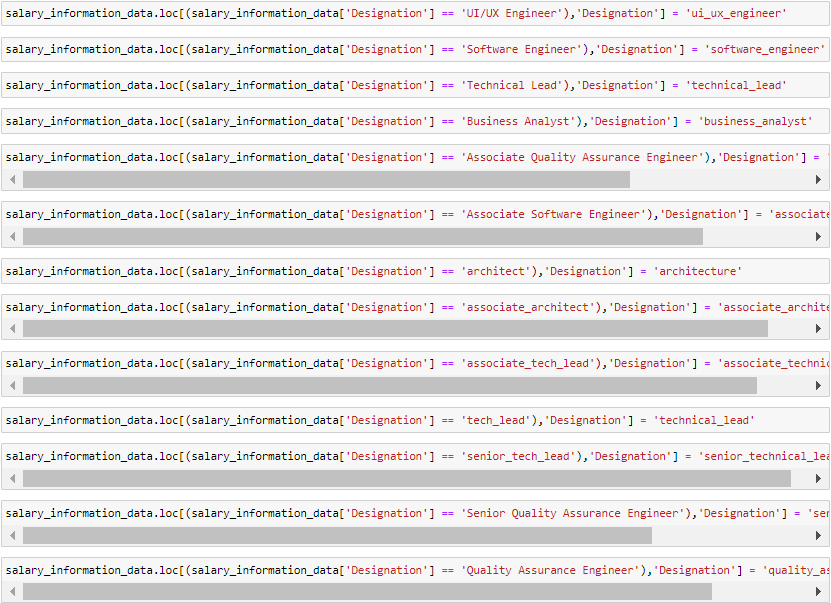
7.3.1.4 Designation preprocess

Figure 19 Designation Preprocess

Whilst checking the dataset there were designation or in other words known as job rolls was in different wording, but represented the same job roll, the above snippet code is representing the conversion to the common labels.

### A picture containing text, screenshot, line, font Description automatically generated7.3.2 Label Encoder

Figure 20 Label Encoding

The above snippet code is that in converts the categorical column to numerical which will help the model to train. Since the model will only be able to recognize numeric values, not with string words.

A screenshot of a computer program

Description automatically generated with low confidence7.3.3 Training the classification model

Figure 21 Classification Model Training

Code snippet represents the training of the classification model, by splitting the dataset into train and test data. Hence which will allow the model to be tested with its metrics. This is very important since the salary recommendation will be given to the end user with the model trained.

### 7.3.4 Content based Filtering.

### A picture containing text, screenshot, font, line Description automatically generated7.3.4.1 TfIdfVectorizer

Figure 22 TFIDFVectorizer

The snippet combines the labels and vectorizers which would create a matrix hence this could be passed into the machine learning model.

### 7.3.4.2 Cosine Similarity

Figure 23 Cosine Similarity

According to the content-based filtering based on the user inputs the best similar match of the salary will be gained by the cosine after the input feature has been vectorized into the matrix.

## 7.4 Implementation of API’s

### A screen shot of a computer program Description automatically generated with low confidence7.4.1 Save salary Information.

Figure 24 Save Salary information.

The above API snippet helps to save the new salary inputs that the user is entering into the system, which accepted as a post API, at the same time the dataset will also be saved by using the same API to make use of the content-based filtering, along with the new data saved into the system.

### A screen shot of a computer program Description automatically generated with low confidence7.4.2 Get Recommendation

Figure 25 Get Recommendation

Get recommendation API will be used to get the salary recommendation for the relevant inputs from the end user.

### A screen shot of a computer program Description automatically generated with low confidence7.4.3 Update rating

Figure 26 Update Rating

The update rating API is a PUT request. It accepts the ID as request parameter from the client, hence will be easy to be queried from the database and update the salary with the new rating that the user has given.

### A screen shot of a computer program Description automatically generated with low confidence7.4.4 Save encoded data.

Figure 27 Save Encoded Data

This API will save the will read the encoded labels CSV and save the information to the database.

## 7.5 Chapter Summary

The Implementation chapter author has explored the technology selection in this chapter, and under the selection of technology frameworks, languages selected IDEs and libraries subjects were covered. The choice of the dataset and how it operates have then been explored. Lastly, the author covered the system's identified essential features by providing code samples and discussing how they were done.

# Chapter 08: Testing

## Chapter Overview

In this chapter the author will be skimming through the testing process that has been involved during the period of implementing the proposed thesis. The author will cover the information about the objectives and goals, functional testing, non-functional testing, and integration testing.

## 8.2Objectives and Goals of Testing

The main objective and goal of the testing process is to make sure that the build software by the author is based on the proposed requirement to validate the following these.

The following testing goals that the author must achieve are as follows:

* Ensuring that the recommendation model is working according and the content-based filtering technique.
* Validating if the code written by the author followed the best practices.
* Making sure that the errors are being handled properly, so that the system wont crash to the end user.
* The system should satisfy the MoSCoW technique pattern mandatory "Must have" and "Should have" of the functional requirements.
* Another goal is to make sure the non-functional requirement is appropriately identified and satisfies the system.

## 8.3 Testing Criteria

The test plan for the Salary recommendation system can be categorized in 2 different categories.

* Functional Testing
* Non-Functional Testing

For each of the prototype's functional and non-functional needs, test cases were written independently to test the application. Hence these test cases will help to determine if the implemented system is functioning as intended.

## 8.4 Model Testing

According to the proposed system by the author, the Salary recommendation system was done in a hybrid approach. The model that was built by the author was a Random Forest Classifier, the other was a most common technique in recommendation system which is the Content-Based filtering approach. Moreover, other classification models were also used by the author to compare what would be the best fit model for the proposed approach.

|  |  |
| --- | --- |
| Model | Testing approach |
| Random Forest Classifier | Accuracy, Precision, Recall, F1 score |
| Support Vector Machine | Accuracy |
| Decision Tree Classifier | Accuracy |
| k Neighbors Classifier | Accuracy |
| Content Based filtering | Cosine Similarities |

Table 26 Model Testing approach

### 8.4.1 Models accuracy

There were different models that were used by the author to make the recommendation system and each model that has been built had given different results as depicted in below diagram Chart

Description automatically generated

Figure 28 Model accuracy chart

As the above diagram were the model accuracy results, which was used to get the best possible salary for the relevant user input given to the model.

|  |  |
| --- | --- |
| Model | Accuracy |
| Random Forest Classifier | 56.5% |
| Support Vector Machine | 5.3% |
| Decision Tree Classifier | 56.4% |
| k Neighbors Classifier | 16.5% |

Table 27 Model Accuracy

### 8.4.2 F1 Score

To evaluate the performance of the model, the F1 score was used. It is a metric that takes int account both precision and recall. The model which was most suitable was the Random Forest Classifier which returned an F1 score of 0.55. This indicates that the model has moderate performance in predicting the target variables.

### 8.4.3 Precision

According to calculations, the precision score, which expresses the percentage of accurate positive predictions among all positive predictions generated by the model, was 0.73. This indicates that 73% of the time, the model's predictions of positive situations were accurate. In applications where false positives are expensive, a high precision score is preferred. The accuracy score in this investigation shows that the model is successful at locating positive cases with a relatively low percentage of false positives.

## 8.5 Benchmarking

The bench marking of the system is that the dataset which has been used to train the classification model and relevant content-based filtering technique was totally based on the Sri Lankan IT industry dataset, there were no foreign records used hence full filling the requirement of the system. In addition, the system builds with the combination of both domains such as solving the social issue which is salary discrimination and few recommendations technique haven’t been implemented according to the study of the author, which is a huge advantage for the IT sector in Sri Lanka to get to know the possibility of salary that everyone would gain. Finally, the system int and suggestion is based only on 4 main features they are Designation, Company size, Year of experience and education level of individual indicating the results of recommendation would be more suitable for the individual.

## 8.6 Functional Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case | FR ID | User Action | Expected Result | Actual Result | Result Status |
| 1 | FR01 | Users can enter salary information details. | All the inputs should be filled else the error message should be shown. | Error message has been shown to the end user. | Passed |
| 2 | FR02 | User inputs data to get the salary recommendation data based.  on the parameters that the user has pooled into the system. | Show the recommendation data for the user input passed. | The most predictable data has been shown to the user | Passed |
| 3 | FR03 | Saving the recommended data based on the user input into the database. | Run on another thread and save the data to the database. | Saved the data to the database by using another thread. | Passed |
| 4 | FR04 | Saving the salary information provided by the information user to the database. | Saving the data to the database. | Saved the data to the database. | Passed |
| 5 | FR05 | Recommending relevant salary data for the user’s entered inputs. | Show result from the model and content-based filtering | Displaying the result relevant to the user input from model and content-based filtering. | Passed |
| 6 | FR06 | The system allows the user to give ratings to the recommended salary data. | Show the edit button for the rating to the user to give a score for the salary that is displayed. | Show the edit button for the rating to the user to give a score for the salary that is displayed. | Passed |
| 7 | FR07 | The system shows error messages for inappropriate use of the system. | Display error messages | Display error messages | Passed |
| 8 | FR11 | Adjusting the relevant salary recommended salary to the latest inflation rate. | Update the salary while saving to the current inflation if the year payment is before the current year. | Update the salary while saving to the current inflation if the year payment is before the current year. | Failed |
| 9 | FR12 | Update the database after the User has given the score. | Updating the rating column for the relevant salary. | Updating the rating column for the relevant salary. | Passed |
| 10 | FR13 | Sort the content-based filtering according to the rating given to the salary | Sorting according to the rating given to the salary. | Sorting according to the rating given to the salary. | Passed |

Table 28 Functional Testing

## 8.7 Module and Integration Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module | Input | Expected Output | Actual Output | Status |
| Input New Salary Information for the system. | Do validation if all the inputs have been entered. | Display error message saying to fill all the important inputs. | Displayed error message saying to fill all the important inputs. | Passed |
| Input fields are entered with the datatype expected. | Display error message informing to enter correct input value. | Displayed error message informing to enter correct input value. | Passed |
| Save the Data after clicking on the Submit button | Display success message and save the data to the database. | Displayed success message and save the data to the database. | Passed |
| Get recommendation for the relevant User inputs. | Do input validation | Display error message | Displayed error message | Passed |
| Input data to get salary recommendation | Show recommended salary for relevant input. | Showed recommended salary for relevant input. |
| Save the recommended data to the database. | Saving to the database with the recommended data. | Saved data to the database |
| Update Score for the Salary | Once clicking on the save button the score should be updated to the relevant salary amount. | Show success message that the rating is updated. | Showed the relevant success message. | Passed |
| Once click on the cancel button the score shouldn’t be updated. | Show success message that the rating is cancelled to updated. | Showed the relevant message. |

Table 29 Module and Integration Testing

## 8.8 Non-Functional Testing

There are few non-functional requirements identified by the author for the proposed system, and which has been depicted in the SRS chapter. According to this chapter the non-functional requirements will be discussed in detail. The accuracy testing of the model is explained under the Model accuracy.

### 8.8.1 Performance Testing

The model performance testing was done by using the F1 score, precision and it was explained in the above chapter. As a result, in this topic the author will be doing showing the performance between the integration to get the recommended salary for the given user input, but however the performance testing was done in local, which could differ if the deployment has done to the cloud.

The diagram below depicts the time scale to get the salary recommendation for the user input that has been passed to the backend server.

Graphical user interface, application, table, Excel

Description automatically generated

Figure 29 Performance Testing One

Graphical user interface

Description automatically generated with low confidenceThe below diagram shows the amount of time that has been taken whilst requesting the backend server to get the recommendation.

Figure 30 Performance Testing Two

### 8.8.2 Security Testing

The security of the system is maintained by not collection information about the user profile, on who is entering the salary and the salary information the user inputs to the system doesn’t save the information of the company that an individual is working on, therefore there has been no violation of collecting personal information that could be used for false interpretation in the future.

### 8.8.3 Usability Testing

The ease with which users can interact and navigate through a user interface is a key indicator of an application's usability. This system was developed with simplicity as a top priority to improve its general use. The screen shot of the application could be referred to in the implementation chapter.

### 8.8.4 Maintainability Testing

The backend, frontend as well as the model implementation have been properly modularized so that it would be very easy to understand and skim through how to rebuild the system above for the future scope of the proposed project.

## 8.9 Limitations of testing process

The limitation of the testing process could be the Non-functional testing. Since the system has not been deployed to cloud, the application will be running in the local, therefore it will show a better performance due to only one user could be able use the application at a time, therefore the responses and all other testing will be very high due to low scalability.

## 8.10 Chapter Summary

In this chapter the author skimmed through the objectives and goals of the testing to the salary recommendation system, moreover the trained model of the system in order to recommend the best recommended salary for user was also testing with different kind of machine learning metrics. The bench marking of the system has also been done based on the dataset that the author has collected along with it the hybrid approach, which is done to solve both the domain issues identified. Finally, the testing of the functionality and non-functionality was also discussed along with the limitation of testing.

# Chapter 09: Evaluation

## 9.1 Chapter Overview

## 9.2 Evaluation Methodology Approach

## 9.3 Evaluation Criteria

## 9.4 Self Evaluation

## 9.5 Selection of Evaluators

## 9.6 Evaluation Results

### 9.6.1 Expert Opinion

### 9.6.1.1 Domain Expert

### 9.6.1.1.1 Concept

### 9.6.1.1.2 Solution

### 9.6.1.2 Technical Expert

### 9.6.1.2.1 Scope

### 9.6.1.2.2 Architecture of the Solution

### 9.6.1.2.3 Implementation of the Solution

### 9.6.2 Focus Group Testing

### 9.6.2.1 Prototype Features

### 9.6.2.2 Usability

## 9.7 Limitations of Evaluation

## 9.8 Evaluation of Functional Requirements

## 9.9 Evaluation of Non-Functional Requirements

## 9.10 Chapter Summary

# Chapter 10: Conclusion

## 10.1 Chapter Overview

This chapter will inform about the entire research by highlighting the aims and objectives that the author has identified during the research process. Moreover, will be discussing the new and improvement skills applied to implement the proposed system along with its learning outcomes, challenges as well as problems faced throughout the period. At the end, according to the author would discuss the limitation, future enhancement could be done to improve this proposed thesis.

## 10.2 Achievements of Research Aims and Objectives

### 10.2.1 Achievements of Aims

*The aim of the research is to design, develop and evaluate a recommendation system that would help individual to beware of the approximate salary that an individual would acquire based on their few characteristics, hence by knowing will be able to overcome salary discrimination because the individual will have some kind of knowledge about the salary they would gain.*

The aim of the research project was successfully achieved by designing, developing as well as evaluating the recommendations architecture to overcome salary discrimination in the organizations and resolve few recommendation problems.

### 10.2.2 Achievements of Objectives

|  |  |  |
| --- | --- | --- |
| Objective | Description | Status |
| Problem Identification | Critically analyzing the problem in domain such as the financial sector as well as the recommendation system problems. | Completed |
| Literature Survey | Critically analyzing the possible similar/existing works accordance to the domain selected | Completed |
| Requirement Elicitation | Discussions and critically analyzing the user requirements and will it be useful in the industry. | Completed |
| Design | Designation of the Salary recommendation system for with UI/UX guidelines | Completed |
| Development | Development of the proposed systems prototype such as the Machine learning, backend server and frontend software | Completed |
| Testing and Evaluation | Testing the prototype if it’s feasible for the end users and evaluating the project with important metrics. | Completed |
| Documenting the progress of the research | Structuring the documentation of the thesis and showing the progress to the supervisor. | Completed |
| Publish finding | Publishing a research paper since a salary recommendation system has not been done, according to the study of the author. | Incomplete |

Table 30 Achievement of Objectives

## 10.3 Utilization of Knowledge from the Course

|  |  |
| --- | --- |
| Module | Description |
| Programming Principle one and two. | This module was a kick start to learn basic programming languages, the author was able to learn about python language and its benefits. |
| Object-oriented Programming (OOP) | This was one of the main concepts and very much helpful to build software in a structured manner both in personal software as well as in the industry. |
| Software Development Group Project (SDGP) | In this module the author was able to make a proper monolithic project that thought on how to build software in an industry level and helped to improve the documentation skills. |
| Database System | With the aid of this module, the author was able to grasp the fundamentals of writing SQL queries and using databases to store data for the project. |
| Algorithms: Theory Design and Implementation | The knowledge gained in this module was very important, since it helped to properly analyze and design algorithms with high performance. |
| Web Design and Development | The author was able to learn basic knowledge and build Frontend Application with UI/UX guidelines by using HTML, CSS as well JavaScript. |

Table 31 Utilization of Knowledge from Course

## 10.4 Use of Existing Skills

Throughout this degree program the author was able to gain several skills which were very useful when developing the proposed system. The gain skilled are listed below:

**Python:** The author had a prior experience in python language since during the SDGP author fully worked in this language. The author learned about the language by watching videos on YouTube as well as using LinkedIn Learning. As result due to this prior knowledge it helped the author to implement the Backend implementation in flask framework.

**Database (NoSql):** These existing skills help the author to make use of database to structure as well as store the new salary informationby using PostgreSQL.

**Industrial placement skills:** The author successfully completed as full-time java backend developer in Digiratina Technology Solution. Also, the author was involved in designing databases and helping the co-worker in implementation process, plus reviewing the codes of co-workers.

## 10.5 Use of New Skills

**Recommendation system** – The author has never worked on a recommendation system or another machine learning module, hence this was an added skill plus an advantage for the author on to learn new scope apart from known knowledge. The author learned recommendation systems by watching videos and doing sample projects with the help of online platforms such as Coursera and Udemy.

**React JS** – Even though the author had a little bit of understanding in JavaScript, due to full time working as a backend end developer during the industrial period, author had to learn react JS as well to implement the Frontend for the proposed system, this was learned through watching YouTube videos and reading the react official documentation.

**Docker** – Docker was learnt by the author to deploy the proposed system both the backend and frontend, but at the end the author was not able to deploy the project, but still the skill was still learned with the help of LinkedIn learning.

**Google Cloud Platform** – Learned a bit on GCP to know how to host project.

## 10.6 Achievement of Learning Outcomes

|  |  |
| --- | --- |
| What has been learned | Learning Outcome |
| This thesis has helped the author to gain knowledge about the domain that has been worked on through the period | LO4 |
| The author developed time management skills by carefully preparing each project phase. | LO2 |
| Building recommendation system with different kind of classification and recommendation techniques was completely new for the author, which must be learnt during the span of the project timeline. | LO5 |
| The proposed project must be done all by the author, hence all the errors and other functionalities should be solved and researched by the author himself, hence problem-solving skills was learned. | LO5, LO7 |
| By continuous research and self-learn the author was able to identify the technologies that should be used to build the proposed system. Technologies to build the Frontend, backend as well as the Machine Learning algorithm. | LO1 |
| The requirement gathering process was also improved, since formal interviews, prototyping as well as survey techniques was used to gather information and validate if this system is feasible and helpful for upcoming individuals in the IT industry. | LO3 |
| Documentation is a one of essential part, since the author should validate the problem and the research domain that this thesis involves in. hence documentation skills was mandatory as was able to improve. | LO8 |

Table 32 Achievement of Learning Outcomes

## 10.7 Problems and Challenges Faced

|  |  |
| --- | --- |
| Challenges | Solution |
| Lacking knowledge in required technologies | Since the author was not aware of the Machine Learning techniques and how to implement, the author had to do many self-learning and do sample projects to implement the proposed system. |
| Time constraint | Even though this was yearlong module, the author had other modules as well to focus, but still with the help of time management skills was able to complete the proposed system successfully. |
| Deployment of the project to GCP. | The author faced deployed issue to GCP cloud, since the database connection using SQL Alchemy didn’t work as expected the backend server wasn’t successfully deployed, but the Frontend was able to deploy using GCP, hence the author decided to use the application in the local host server. |

Table 33 Problems and Challenges Faced

## 10.8 Deviations

The initial goal of the project was also introduce reinforcement learning, since the salary of an individual will be changing over the period of time, hence based on the system whenever the new salary is added the author planned to use reinforcement leaning which would improve the Salary recommendation for the target audience, but due to the time constraints and the effort required to learn about on how to build reinforcement for the particular was not feasible.

Moreover, the author also wanted to use Deep learning to build the recommendation system, due to lack of dataset for Sri Lankan IT industry salary information, was not able train a Deep learning technology hence, used Machine learning as well as recommendation technique to build the system.

In addition, the author also planned to do prediction for future salary, but due to no dataset availability the author was not able to build prediction for the future salary as well.

## 10.9 Limitations of Research

* **The scope of the project** – The scope of the project was only to build the salary recommendation for Sri Lankan IT industry individual, which could be expanded globally and to different organization sector. The author only considered a few input features which could be expanded based on the geographical locations.
* **System Application** – The system application was designed and implemented only as a website. It would not support mobiles since it has not been responsive to be opened in mobile phones due to limitation of the time to build the project.
* The system is only limited for few Currencies input which would not accept all the possible currency rates across the globe and the output of the Salary will be recommended in Sri Lankan rupees, not in any other salary.

## 10.10 Future Enhancements

* Improve the accuracy of the system by using better algorithms, since the author got only 56% maximum with RandomForestClassifier.
* Do reinforcement learning so that the salary will be up to date, since salary would change over a period.
* Work on Salary prediction as well, hence the users will be helpful to make a future forecast.
* Since the salary dataset is only based on Sri Lankan region, it could be diversified across other regions.
* Making sure that the application is available on all sorts of devices, both on websites as well as mobile applications.
* Accepting more input features from the end user such as Reviews about the recommended salary and accepting location or state for better recommendation of salary.

## 10.11 Achievements of the contribution to body of knowledge

According to the author’s research gap, author tried to solve to build a hybrid recommendation system to solve few recommendation problems such as cold start, lack of data problem and many more. And the author wanted to solve the salary discrimination in between employees within an organization in the IT industry, hence author build a hybrid abroad with the help of combining the RandomForestClasssfier and Content based filtering approach to solve the domains that has been selected. Therefore, with this approach and novel implementation the individual will be able to bargain on the salary if they are biased, which would overcome a bit of discrimination in the IT industry market.

In addition to the machine learning model the contribution will be the hybrid system with the Sri Lankan specific dataset of the people in IT industry which has never build by any individual with the combination of these domains.

## 10.12 Conclusion Remarks

In this chapter the author concluded the research thesis by evaluating the research gap from getting feedback from the domain and technical expertise as well as by allowing the end users to use the system, and how well it has helped. Moreover, the author also gained new skills and improved the existing skills by implementing the proposed system. In fact, during the time of doing the research thesis, the author faced a few problems, limitations and identifying a proper research gap, hence learnt on how well it must be properly planned and discussed to overcome the problems. Finally, the author was able to gain experience and knowledge both domain wise and technical wise by the help of this project.

# REFERENCE

# APENDIX A - User and Item based filtering Diagram.

## 

A screenshot of a diagram

Description automatically generated with low confidence

Figure 31 User and Item based filtering (Wu, 2022)

# APENDIX B – Decision Tree Structure

A diagram of a decision tree

Description automatically generated with medium confidence

Figure 32 Decision Tree Structure (Bansal, Goyal and Choudhary, 2022)

# APPENDIX C - Gant Chart

Chart, bar chart

Description automatically generated