AN AUTOMATED SYSTEM FOR CLASSIFYING CONFERENCE PAPERS

SEON NGAN CHOON HAN

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Hons.) Software Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2021

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : 

Name : Seon Ngan Choon Han

ID No. : 1706067

Date : 15/04/2021

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"AN AUTOMATED SYSTEM FOR CLASSIFYING CONFERENCE PAPERS"** was prepared by **SEON NGAN CHOON HAN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : kckhor

Supervisor : Khor Kok Chin

Date : 4/5/2021

Signature :

Co-Supervisor :

Date :

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my utmost gratitude to my research supervisor, Dr. Khor Kok Chin for his invaluable advice, guidance and enormous patience throughout the project's research and development.

In addition, I would also like to express my gratitude to my family members for their support throughout the project as well my friends who had helped and given me encouragement.

ABSTRACT

In the research conference domain, paper assignment process often poses as a time-consuming and repetitive task for a chairman. A chairman is required to manually review the contents of a research paper, before assigning it to a suitable reviewer. This project is aimed to develop an automated web-based conference paper system for the manual process of assigning papers to reviewers by using classification models. The project is also aimed to select the best classification model for the system, based on an empirical study. The Knowledge Discovery in Databases (KDD) process was followed as a formal data mining methodology where 1000 AI conference papers were carefully collected, pre-processed and transformed to numerical representations through TF-IDF vectorisation. A randomised stratified 5-fold cross validation was then applied on several data mining algorithms and evaluated using the F-measure as a metric. The Support Vector Machine algorithm resulted in the highest F-measure (0.906), followed closely by Logistic Regression (0.903), Random Forest (0.891), Naïve Bayes (0.880), K-Nearest Neighbour (0.831) and lastly, Decision Tree (0.778). Grid search optimisation was then performed but no significant improvements could be observed. The best classification model was then deployed to a web-based research conference system. The web-based system was developed using the Django web framework, based on a system architecture defined in this project called the Enhanced 3-Tier Web-based System with a Data Mining Layer. In conclusion, an automated paper classification system was successfully developed using classification models and its practical usage was demonstrated on a web-based research conference system to help chairmen in assigning papers to suitable reviewers.

TABLE OF CONTENTS

DECLARATION **ii**
APPROVAL FOR SUBMISSION **iii**
ACKNOWLEDGEMENTS **v**
ABSTRACT **vi**
TABLE OF CONTENTS **vii**
LIST OF TABLES **xi**
LIST OF FIGURES **xiii**
LIST OF SYMBOLS / ABBREVIATIONS **xv**
LIST OF APPENDICES **xvi**

CHAPTER

1 INTRODUCTION **17**
1.1 Introduction 17
1.2 Background of the Problem 17
1.3 Problem Statement 18
1.3.1 Human Dependent Task of Paper Review Process 18
1.3.2 Repetitive Task of Paper Review is Time Consuming 18
1.4 Project Objectives 19
1.5 Proposed Solution 19
1.5.1 User Flow Chart 19
1.5.2 An Enhanced 3-Tier Web-based System with Data Mining Layer 21
1.6 Proposed Approach 22
1.6.1 Agile Methodology 22
1.7 Project Scope 23
1.7.1 Web-based System 23

1.7.2 Target User and Account 23
1.7.3 Data Mining Modelling 24
1.7.4 Programming Tools 24

2 LITERATURE REVIEW 25
2.1 Reviewing Modules of Similar System Based on a Design Framework 25
2.2 Data Mining Workflow and Algorithms in Document Classification 28
2.3 Existing Conference Paper Classification Systems 33
2.4 Agile Software Development Methodology 35

3 PROJECT METHODOLOGY AND PLANNING 37
3.1 Introduction 37
3.1.1 Agile Software Development Methodology 37
3.1.2 Feature Driven Development 37
3.2 Data Mining Methodology 39
3.2.1 Knowledge Discovery in Databases (KDD) process 39
3.3 Low-Fidelity Design 41
3.4 Project Plan 41
3.4.1 Work Breakdown Structure 41
3.4.2 Gantt Chart 41
3.5 Development Tools 41
3.5.1 Programming Language 41
3.5.2 Frameworks 42
3.5.3 Database 43
3.5.4 Integrated Development Environment 43

4 PROJECT SPECIFICATION 44
4.1 Introduction 44
4.2 Use Case Diagram 44
4.3 Use Case Descriptions 45

	4.4	Software Requirements	54
		4.4.1 Functional Requirements	54
		4.4.2 Non-functional Requirements	57
5	**SYSTEM DESIGN**		**58**
	5.1	Introduction	58
	5.2	System Architecture Design	58
		5.2.1 Presentation Layer	59
		5.2.2 Application and Persistent Storage Layers	59
		5.2.3 Data Mining Layer	60
	5.3	Database Design	61
	5.4	Web Application User Interface	62
		5.4.1 Common views	62
		5.4.2 Chairman’s views	63
		5.4.3 Author’s views	66
		5.4.4 Reviewer’s views	68
6	**PROJECT IMPLEMENTATION**		**70**
	6.1	KDD Process Implementation	70
		6.1.1 Selection	70
		6.1.2 Pre-processing	72
		6.1.3 Transformation	72
		6.1.4 Data Mining	73
		6.1.5 Evaluation and Results	74
		6.1.6 Optimisation and Results	75
	6.2	Web Application Implementation with Django	81
		6.2.1 Model	81
		6.2.2 View	83
		6.2.3 Template	87
7	**TESTING**		**88**
	7.1	Unit Testing	88
		7.1.1 Admin Login Module	88

7.1.2 Account Creation by Admin Module 88
7.1.3 User Login Module 90
7.1.4 Create Conference Module 90
7.1.5 Invite Reviewers Module 91
7.1.6 Publish Reviews Module 91
7.1.7 Assign Reviewers Module 92
7.1.8 Download PDF Module 92
7.1.9 Chair Decision Module 92
7.1.10 Paper Submission Module 93
7.1.11 Camera-ready Submission Module 93
7.1.12 Update Domain Expertise Module 94
7.1.13 Give Review Module 94
7.2 Integration Testing 95
7.2.1 Account Creation by Admin and User Login 95
7.2.2 Create Conference and Paper Submission 95
7.2.3 Invite Reviewer and Assign Reviewer 96
7.2.4 Give Review and Submit Chair Decision 96
7.2.5 Camera-ready Submission and Download PDF 97
7.3 System Testing 98
7.3.1 Paper Assignment Process 98
7.3.2 Paper Review Process 99

8 CONCLUSION AND RECOMMENDATION 101
8.1 Introduction 101
8.2 Benefits of the Project 101
8.3 Limitations of the Project 101
8.4 Recommendations for Future Improvements 102
8.5 Conclusion 102

REFERENCES 103

APPENDICES 107

LIST OF TABLES

Table 2.1: Comparison of modules in OpenConf web-based system and this project 27

Table 2.2: Comparison of NB and SVM 30

Table 2.3: Characteristics and Disadvantages of Data Mining Algorithms 30

Table 2.4: Differences between traditional and agile approach 35

Table 3.1: Description of each activity in FDD process 38

Table 4.1: Upload Paper use case description 45

Table 4.2: Check Paper Status use case description 46

Table 4.3: Accept Chairman Invitation use case description 47

Table 4.4: Give Review use case description 48

Table 4.5: Invite Reviewer use case description 49

Table 4.6: View Papers use case description 50

Table 4.7: Assign Paper to Reviewer use case description 51

Table 4.8: Recommend Reviewer use case description 52

Table 4.9: Check Review by Reviewer use case description 53

Table 6.1: Best parameters of each model 78

Table 6.2: Models description 81

Table 6.3: Views description 83

Table 6.4: Templates description 87

Table 7.1: Unit test for Admin Login module 88

Table 7.2: Unit test for Account Creation by Admin module 88

Table 7.3: Unit test for User Login module 90

Table 7.4: Unit test for Create Conference module 90

Table 7.5: Unit test for Invite Reviewers module 91

Table 7.6: Unit test for Publish Reviews module 91

Table 7.7: Unit test for Publish Reviews module 92

Table 7.8: Unit test for Download PDF module 92

Table 7.9: Unit test for Chair Decision module 92

Table 7.10: Unit test for Paper Submission module 93

Table 7.11: Unit test for Camera-ready Submission module 93

Table 7.12: Unit test for Update Domain Expertise module 94

Table 7.13: Unit test for Give Review module 94

Table 7.14: Integration test for Account Creation by Admin and User Login modules 95

Table 7.15: Integration test for Create Conference and Paper Submission modules 95

Table 7.16: Integration test for Invite Reviewer and Assign Reviewer modules 96

Table 7.17: Integration test for Give and Submit Chair Decision modules 96

Table 7.18: Integration test for Camera-ready Submission and Download PDF modules 97

Table 7.19: System test for paper assignment process. 98

Table 7.20: System test for paper review process. 99

LIST OF FIGURES

Figure 1.1: User Flow Chart 19

Figure 1.2: System Architecture Diagram 21

Figure 1.3: Iterative Process of Agile Methodology 22

Figure 2.1: Homepage of openconf.com 25

Figure 2.2: Reviewer page of openconf.com 26

Figure 2.3: Chairman Review page of openconf.com 26

Figure 3.1: Feature Driven Development Process 37

Figure 3.2: KDD process 39

Figure 4.1: Use Case Diagram of the web-based system 44

Figure 5.1: An Enhanced 3-Tier Web-based System with Data Mining Layer 58

Figure 5.2: Entity Relationship Diagram 61

Figure 5.3: Login page 62

Figure 5.4: Conferences 62

Figure 5.5: Conference creation 63

Figure 5.6: Reviewers invitation 63

Figure 5.7: Paper submissions under a conference 64

Figure 5.8: Paper details and reviews 64

Figure 5.9: Paper assignment 65

Figure 5.10: Paper submission 66

Figure 5.11: Submissions of an author under a conference 66

Figure 5.12: All submissions of an author 67

Figure 5.13: Paper details and reviews 67

Figure 5.14: Update domain expertise 68

Figure 5.15: Assigned papers 68

Figure 5.16: Paper details and reviews 69

Figure 5.17: Giving a review 69

Figure 6.1: Arxiv open-source journals by Cornell University 70

Figure 6.2: Check paper for conference proceedings under a publisher 71

Figure 6.3: Conference papers distribution among classes 71

Figure 6.4: Symbols removal 72

Figure 6.5: Stop words removal and lower casing 72

Figure 6.6: Applying TF-IDF vectorisation 72

Figure 6.7: Sample TF-IDF values of a paper 73

Figure 6.8: Cross-validation of each algorithm 74

Figure 6.9: F-Measure of each data mining algorithm 74

Figure 6.10: Optimisation of the SVM model through grid search 77

Figure 6.11: Optimisation of the LR model using Elastic-Net through grid search 77

Figure 6.12: Optimisation of the LR model using Lasso (L1) and Ridge (L2) through grid search 78

Figure 6.13: Optimisation of the RF model through grid search 78

Figure 6.14: Comparison of F-measure before and after grid search optimisation 80

Figure 6.15: Serialising the TF-IDF vectoriser and model instances into a file 80

Figure 6.16: De-serialising the TF-IDF vectoriser and model instances 81

LIST OF SYMBOLS / ABBREVIATIONS

ACM	Association of Computing Machinery
AI	Artificial Intelligence
FDD	Feature Driven Development
GUI	Graphical User Interface
HTML	Hypertext Mark-up Language
IDE	Integrated Development Environment
ISI	Institute for Scientific Information
JUCS	Journal of Universal Computer Science
KDD	Knowledge Discovery in Databases
KNN	K-Nearest Neighbour
MVT	Model-View-Template
NB	Naïve Bayes
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
RDBMS	Relational Database Management System
SQL	Standard Query Language
SVM	Support Vector Machine
TF – IDF	Term Frequency – Inverse Document Frequency
UI	User Interface
XP	Extreme Programming

LIST OF APPENDICES

APPENDIX A: Login page 107

APPENDIX B: Chairman user page 107

APPENDIX C: Reviewer user page 108

APPENDIX D: Author user page 108

APPENDIX E: Work Breakdown Structure 109

APPENDIX F: Gantt Chart 110

CHAPTER 1
INTRODUCTION

1.1 Introduction

This chapter aims to provide a preliminary structure for the whole project. The content to discuss is the background of the problem which provides a general view of the conference paper review domain. The problem statement section shall describe the reality of the problem and the need to solve the problem.

The goals and objectives that this project aims to achieve are also outlined in the project objectives section. The proposed solution provides a system overview of how the system shall be used and how it should be designed. Lastly, the proposed approach defines the software development methodology that this project aims to use to leverage a quality software development lifecycle to this project.

1.2 Background of the Problem

According to Nguyen et al. (2016), assigning papers to reviewers by the reviewing chair is an important task for conference chairs and the committees which requires a precise matching between reviewers and papers. It is also mentioned that the reviewing task must be made within days after the submission due date, which puts a stressful burden on the conference chairs.

Not only reviewing chairs have to assign the papers to the reviewers, but they also have other tasks at hand, such as conducting discussions with reviewers, logging of papers, managing conference schedules, reviewing reviewed papers and also notifying authors of acceptance or rejection of papers.

Looking at the tasks of the reviewing chairs at hand, the amount of work is quite extensive and manual. Hence, having a solution that can automate some of the processes and even the decision made by the reviewing chair can make the paper review process much more efficient and effective. This leads to the establishment of this project which is to solve the problems of reviewing chairs by building an automated system for classifying conference papers.

1.3 Problem Statement

In the research paper domain, one of the research conference chairman's tasks is to assign research papers to reviewers according to their field of topics accordingly. This is a workflow in the research paper domain called paper review and paper selection.

1.3.1 Human Dependent Task of Paper Review Process

The workflow of paper review and paper selection presently is a manual process. Meaning, the task of reading the content of the papers, understanding the topic of the papers and then allocating the papers to the respective topic reviewers and selecting the paper for the conference are all human dependent tasks.

For example, consider a conference about Artificial Intelligence. There may be various sessions in the conference focusing on specific topics such as Machine Learning, Deep Learning, Computer Vision and even AI Ethics. The reviewing chair will need to go through all the submitted papers and categorise the papers according to the conference's sessions.

1.3.2 Repetitive Task of Paper Review is Time Consuming

The problem with this human dependent task is that it is time-consuming, as well as a repetitive task. Therefore, there comes a need to find a solution that is more effective and efficient by automating the current process of paper review. This will help the reviewing chairman to save a lot of time from the repetitive task of finding out the topics of the paper.

Hence, it is proposed that building an automated system which incorporates Data Mining techniques help reviewing chairs to classify the papers automatically. This way, it saves time by knowing the topic right away to decide which reviewer to assign the papers for reviewing. This will not only be preventing possible human errors in the current human dependent task, but as well as saving the time of the reviewing chairs for more important tasks.

1.4 Project Objectives

1. To develop an automated web-based conference paper system for researchers.
2. To automate the manual process of assigning papers to reviewers by using classification models.
3. To select the best classification model for the conference paper system based on an empirical study.

1.5 Proposed Solution

1.5.1 User Flow Chart

The proposed solution for this project is defined with two overviews. Firstly, it is the system overview in terms of the business logic of the system. It is visualised through a user flow chart which clearly defines the business logic of the system.

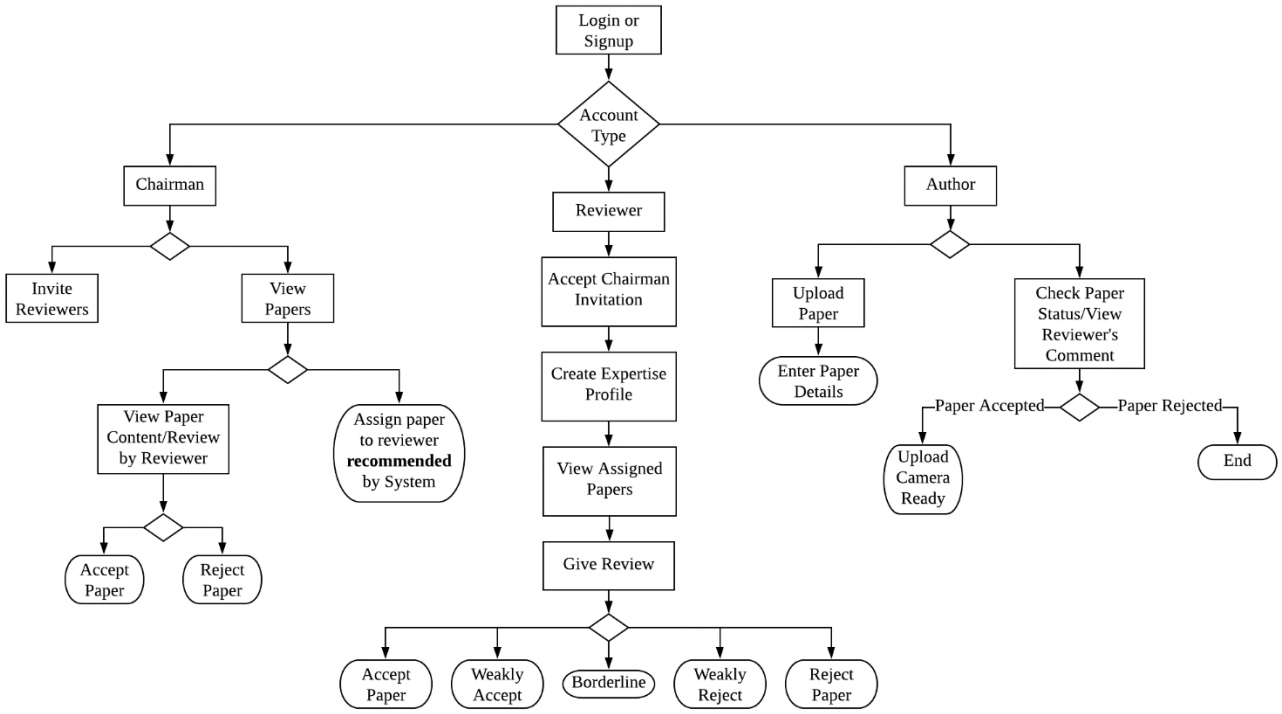


Figure 1.1: User Flow Chart

- **Signup/Login**

The system firstly will allow account creation where the account type should be either chairman, reviewer or author with each account having its business logic.

- **Author**

A user with an author account will be able to upload a conference paper and enter the paper details such as title, abstract and keywords for the paper to be reviewed by the reviewer and chairman. The author can then check paper status where a reviewer's comment on the paper can be seen if the paper has been reviewed. If the paper is accepted, the author can proceed to upload the camera-ready. If the paper is rejected, it is the end process for that paper

- **Reviewer**

A user with a reviewer account will be able to firstly accept chairman's invitation to review papers. After accepting, the reviewer can create an expertise profile where the topic domain to be reviewed by the reviewer is defined. The reviewer can now view papers assigned by the chairman and give a comment and review consist of the option Accept, Weakly Accept, Borderline, Weakly Reject and Reject.

- **Chairman**

A user with a chairman account will be able to choose to invite reviewers first or view papers. In the view papers process, the chairman can view the paper's content along with the reviewer's comments if it has been reviewed and then decide to accept or reject the paper for the conference. Under the view papers process, the system will also match the topic of the paper to relevant reviewers and recommend the chairman to assign the paper one of the reviewers for paper review.

1.5.2 An Enhanced 3-Tier Web-based System with Data Mining Layer

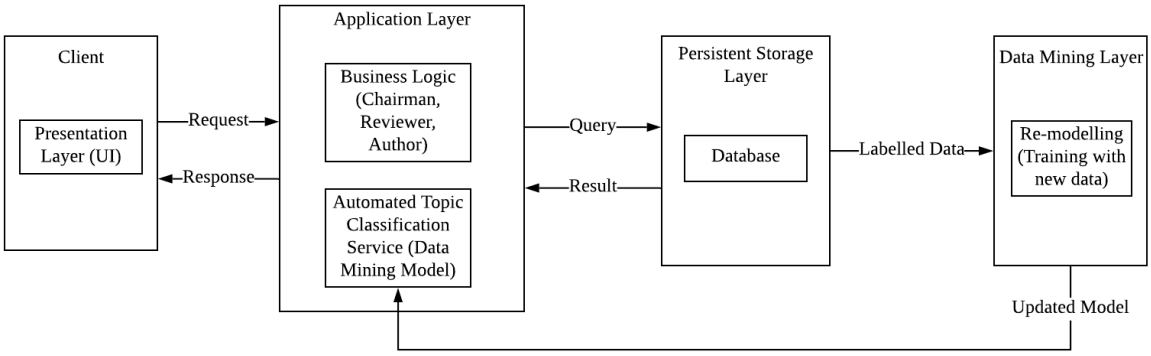


Figure 1.2: System Architecture Diagram

The second system overview is the system architecture. The project will be developed as a web-based system consisting of a 3-Tier Architecture with an additional Data Mining Layer.

- **Presentation Layer**

This layer resides with the Client. This layer is where the Graphical User Interface (GUI) is rendered at the Client's side through a web browser. The user will interact with the client to send requests to the Application Layer where the server resides and receive responses that will be displayed to the user.

- **Application Layer**

This layer resides with the Server. This layer is where the Business Logic such as processes of author, reviewer and chairman accounts are performed upon the request from the Client. The Data Mining Model is also deployed as one of the business logics which is the Automated Topic Classification Service where the best model is derived from the Data Mining Layer. The Application Layer will send queries to the Persistent Storage Layer where the database server is residing and receive results accordingly.

- **Data Mining Layer**

This layer is where the process of re-building the data mining model can be done. Occasionally, when more new data is collected in the database for the conference papers, the data can be queried and send to the Data Mining Layer to re-train the model with new data. This ensures that the model can continuously be improved and also adjust to any changes between topic and keyword pattern.

1.6 Proposed Approach

The proposed approach for this project is by using the Agile methodology.

1.6.1 Agile Methodology

Instead of traditional software development methodologies, the agile methodology will be used which focuses less on documentation, but rather the focus is placed on the communication and code. For example, Extreme Programming (XP) is one of the agile methodologies which is based on the values of simplicity, communication, feedback and courage (Eberlein, 2003).

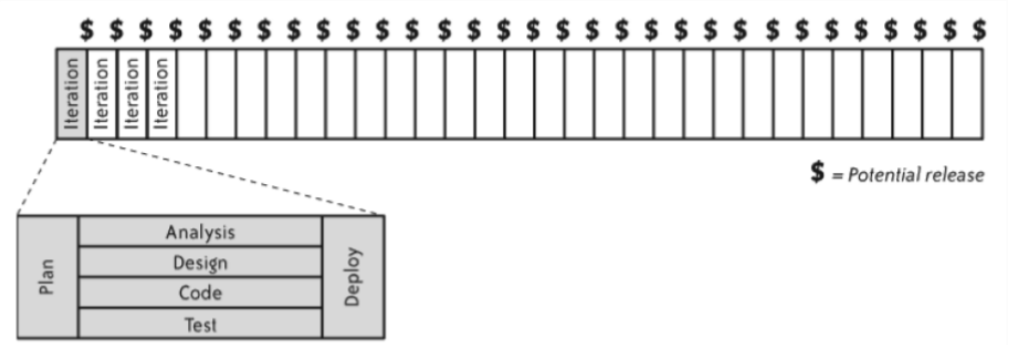


Figure 1.3: Iterative Process of Agile Methodology

In Extreme Programming, the software development lifecycle is iterative and kept in short lengths, referring to Figure 1.2. It is because XP highly values the importance of face-to-face communication and collaboration. Face-to-face collaboration is so impactful that, software developments no longer require separate phases which allows all phases to be worked on together in one iteration (James, 2007).

According to the book written by James (2007), the iterations are focused on completing the stories which are the informal requirements gathered from the relevant stakeholder. For each iteration, all commitment will be put into picking a few stories to be completed. At the end of the iteration, a review is done to the deployed product, whether internally or by the customer stakeholder.

Therefore, in this project, the software development lifecycle which consists of planning, analysis, design, implementation and deployment will be conducted simultaneously for iterations that last a few weeks. These simultaneous activities will be done in a small portion to satisfy the stories gathered from the user. In each iteration, a few of the stories will be picked to be tasked and completed.

1.7 Project Scope

In this section, the project scope will be clearly defined as the following:

1.7.1 Web-based System

The project will be built as a web-based conference system for reviewing papers. The system is mainly targeted for conference paper reviewing chair and authors. The system will allow users to sign up and log in as a chairman, reviewer or author account. Each of the accounts will have their functionalities.

1.7.2 Target User and Account

The system will allow users with the chairman account to have the functionalities of viewing papers, selecting papers to view if there are any comments given by reviewers and selecting to Accept or Reject a paper. The system will also allow a chairman to invite reviewers to start reviewing papers. A chairman will also have the functionality of receiving the system’s recommendation of which reviewer to assign the papers to.

The system will allow a reviewer to use the Accept Invitation functionality when the chairman invites the Reviewer to review the papers. Upon accepting the invitation, the reviewer can create an expertise profile to fill in topics of expertise. The profile will also be used by the chairman or the system to assign relevant papers to the reviewer. The reviewer can then view the assigned papers and provide reviews. A decision on each paper (Accept, Weakly Accept, Borderline, Weakly Reject or Reject), will then be made.

The system will allow users with the author account to use the functionality of uploading paper and entering the paper's details such as title, abstract and keyword. Authors will also have the functionality to check the reviews and comments given by the assigned reviewers and also to check whether the paper has been accepted or rejected. Authors will be able to upload the camera-ready if their papers are accepted.

1.7.3 Data Mining Modelling

The system will have a feature to recommend reviewers to the chairman to review the papers. This feature is called the automated topic classification service for this system. It will be implemented using data mining algorithms that will be chosen by empirical studies. The data sets that will be used are formed based on papers extracted from the publishers such as Springer, ScienceDirect, IEEEXplore and ACM (Scopus and ISI Indexed).

1.7.4 Programming Tools

The system will be developed fully using Python scripting language. Python libraries for machine learning, web application development and database management will be used.

CHAPTER 2
LITERATURE REVIEW

2.1 Reviewing Modules of Similar System Based on a Design Framework

A framework for conference management system was designed by Gupta et al. (2013). The framework outlines the standard modules that a conference paper management system should have, which are:

- Login module for authors, reviewers and chairmen
- Module to add conferences by chairmen
- Module to add comments to papers by reviewers and chairmen
- Module to list comments given by reviewers
- Module to assign papers to reviewers by chairmen

A similar system is used to determine whether the modules that are implemented satisfies the standard modules outlined in the framework. The system that was chosen is also a web-based conference paper system (*OpenConf*, 2020). The modules that are implemented in the system can be seen with the following figures.

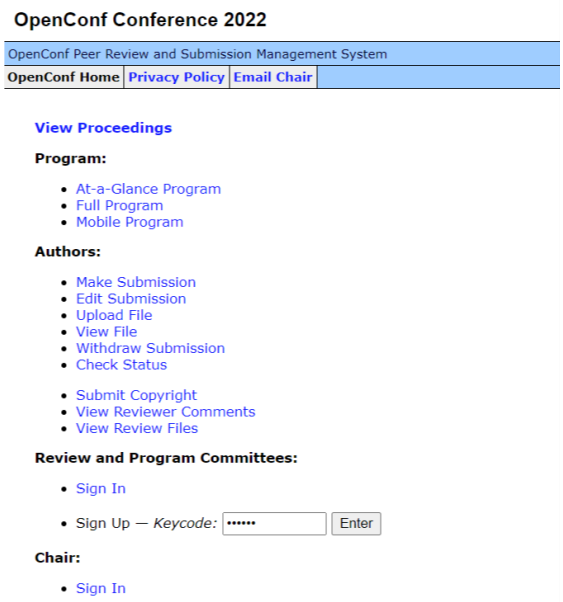


Figure 2.1: Homepage of openconf.com

As seen in Figure 2.1, the system satisfies the login module. However, it does not have a login module for authors.

Submissions to Review:

A blank review form (that opens in a separate window) is available for you to print out if you prefer writing it out before typing it in.

Legend: o *Review completed* x *Review not yet completed*

	Title - click for review form	Abstract	Type	Review Paper
o	1 - Man-Computer Symbiosis		Paper	(53KB)
o	2 - The Computer as a Communication Device		Paper	(53KB)
o	3 - As We May Think		Paper	(53KB)
o	6 - Information Flow in Large Communication Nets		Paper	(53KB)
o	9 - Toward A Cooperative Network Of Time-Shared Computers		Paper	(53KB)
o	11 - Communication networks to serve rapid-response computers		Paper	(53KB)
o	21 - Standard for the transmission of IP datagrams on avian carriers		Paper	(53KB)
o	26 - The Roman Standards Process -- Revision III		Paper	(53KB)
o	27 - The Infinite Monkey Protocol Suite (IMPS)		Paper	(53KB)
o	28 - Electricity over IP		Paper	(53KB)
				ZIP

Figure 2.2: Reviewer page of openconf.com

Figure 2.2 shows the reviewer's page which allows the reviewer to click one of the papers to start reviewing the papers assigned. This satisfies the framework's standard which is to have the module to add comments to papers by the reviewer.

Links:
R-ID – Show review
Reviewer – Show Reviewer info
Submission – Show Submission info

Legend:
Review Status: Marked as Complete Started Not Yet Saved

Recommendation: (1) Reject (2) Probable Reject (3) Marginal Tend to Reject (4) Marginal Tend to Accept (5) Clear Accept (6) Must Accept

Unassign Reviews

S-ID	Submission	Recom.	R-ID	Reviewer	
1	Man-Computer Symbiosis	5	2	Alan Turing	
		5	6	Leonhard Euler	
		6	14	Joseph Marie Jacquard	
		5	15	Edsger Dijkstra	
2	The Computer as a Communication Device	3	2	Alan Turing	
		3	6	Leonhard Euler	
		3	14	Joseph Marie Jacquard	
		2	15	Edsger Dijkstra	

Figure 2.3: Chairman Review page of openconf.com

Figure 2.3 shows the review list of the chairman's page. This page allows reviewer to see the rating given by reviewers and as well as assignment of reviewers to the papers. This page satisfies the framework for modules to list comments given by reviewers and assign papers to reviewers.

The demo of the OpenConf web-based system was further explored, and a summary of modules is produced in Table 2.1 below, which provides a comparison on the modules that this project will cover and the modules that the OpenConf system contains, based on the framework for conference management system by Gupta et al. (2013).

Table 2.1: Comparison of modules in OpenConf web-based system and this project

Modules based on the standard framework	**OpenConf web-based system**	**Modules for this project**
Login for authors, reviewers and chairmen	Only Reviewer and Chairman	Yes
Add conferences by chairmen	Yes	Yes
Add comments to papers by reviewers and chairmen	Yes	Yes
List comments given by reviewers	Yes	Yes
Assign paper to reviewers by chairmen	Yes	Yes
Additional modules from the proposed solution		
Assign paper to reviewers recommended by the system	No	Yes
Upload camera-ready by authors	No	Yes

Referring to Table 2.1, the modules based on the framework are all covered by the OpenConf web-based system. However, the login module is only implemented for reviewers and chairmen. The absence of this module may have negative implications, such as security and management issues (Dehnad, 1989). Exploiters may create bots to flood the website with fake submissions. It is also difficult for the conference system and chairmen to manage the submissions made by authors without credentials. Hence, this project will overcome the limitation by implementing a login module for all types of users.

On top of the modules covered based on the standard framework for conference paper management system, one of the additional modules that this project will implement is to assign papers to reviewers recommended by the system which will be used by the chairman users. This module relates back to the objective of automating the assignment of reviewers using classification models.

There are many motivations to highlight the importance of this module. Firstly, the volume of papers in conferences nowadays routinely reach up to thousands of papers which brings to the need of automating the assigning process which also reduces the time taken to assign the papers, and secondly, automated assignment of reviewers eliminates the bias of the chairmen to select the usual reviewers rather than the most suitable reviewers (Charlin & Zemel, 2013).

The other module that will be implemented in this project that is not covered in the OpenConf system is the module for the author users to upload their camera-ready paper. A camera-ready paper is the final version of the paper that will be published in the conference, where it cannot be revised once it is submitted ("Instructions for the Preparation of a Camera-Ready Manuscript," n.d.). Implementation of all these modules in this project will ensure that the management of conference papers can be digitalised end-to-end and most importantly achieving automation in the paper assignment process.

2.2 Data Mining Workflow and Algorithms in Document Classification

According to Vijayan et al. (2017), the stages of data mining workflow to perform document classification consists of text pre-processing, encoding, dimensionality reduction, classification and performance evaluation.

In text pre-processing, tokenisation is first performed where documents are split into tokens which are either a single word or a phrase. Text pre-processing techniques are then applied to these tokens such as removing stop words and stemming. Stop word removal is to remove frequent words that do not carry any context. Stemming is to reduce all the tokens into their root word. Symbols are also removed from the documents. These techniques will reduce the "noises" in the data

and improve the performance of the classification model as supported by the study done by Uysal & Gunal (2014), where an appropriate set of pre-processing techniques improved the accuracy of the model.

After text pre-processing, the next step is to perform the encoding of the words into vector form. A form of encoding which also utilises weightage of word frequency across documents is the Term Frequency – Inverse Document Frequency (TF – IDF) method. This method allocates a higher weight for a word if the frequency of the word (Term Frequency) appearing in a specific document is high and lesser weight if the frequency of documents (Document Frequency) in which the word appears in is high (Vijayan et al., 2017). However, a limitation of this encoding is that it is a frequency-based encoding that ignores the semantics of each word.

In dimensionality reduction, features are reduced by either removal or selection. Since the features in this project will be the TF-IDF values for all the words in the documents, reduction is performed by selecting only top words with the highest weights to be used as features (Sergienko et al., 2016). It was found that performing dimensionality reduction significantly improves the efficiency of the classification model as lesser features means lesser computation is required, thus reducing the time and memory usage (Sergienko et al., 2016). It is also found that sometimes, the accuracy of prediction models may also be improved through dimensionality reduction as noises in the data is reduced (Vijayan et al., 2017).

As for the classification modelling, a multi-class sentiment classification modelling was performed using various algorithms and it was found that Support Vector Machine (SVM) was the best performing algorithm in terms of accuracy while Naïve Bayes (NB) lagged a little behind SVM (Liu et al., 2017). However, the average execution time was also compared, and it was found that SVM took much longer than NB. These results can be supported by another study, which identifies the advantage and disadvantages of SVM and NB as illustrated in Table 2.2 below.

Table 2.2: Comparison of NB and SVM (Vijayan et al., 2017)

Algorithm	**Advantages**	**Disadvantages**
Naïve Bayes	Simple and efficient	Feature independence is assumed
Support Vector Machine	Performs well in high dimensional data and non-linear boundaries	High time and space complexity

The higher accuracy of SVM can be supported by the ability of the algorithm to perform in non-linear boundaries while the slower execution time is probably due to its disadvantage of high time and space complexity.

On top of that, Table 2.3 also illustrates other possible data mining algorithms that are widely used in the area of text document classification.

Table 2.3: Characteristics and Disadvantages of Data Mining Algorithms

Algorithm	**Characteristic**	**Disadvantages**
Logistic Regression	Linear decision boundary predicting probabilities	Assumption that each data point is independent
K-Nearest Neighbour	Non-parametric classification using distance metric	Computation highly affected by large datasets due to calculating distance for every data point
Decision Tree	Hierarchical decomposition of the dataset based on the labelled data points	Sensitive to noise in the data as the decision tree is completely composed from dataset
Random Forest	Generating random decision trees and uses ensemble learning method to perform voting on the	Slow to make predictions due to comparison computations when high number of trees is used

	best tree	
Neural Networks	Multi-connection of layers where every layer receives the previous layer as input upon applying activation functions	Requires a deep understanding to be able to interpret and tune the complexity of the model according to required use cases

Referring to Table 2.3, Logistic Regression is a classifier with linear decision boundary that predicts probabilities of classes which then can be turned into binary classes of 0 and 1 through the sigmoid function (Fan et al., 2014; Juan & Vidal, 2002). This classifier usually performs well with the assumption that all data points are independent (Huang, 2015). However, according to Kowsari et al. (2019), since text documents may contain topics that are interrelated between different data points, this classifier may not be suitable for text document classification.

K-Nearest Neighbour (KNN) is an algorithm that does not require any parameters for classification where it only requires the number of neighbouring data points and classification is done by calculating the similarity of the data points with each other using distance metrics (Kowsari et al., 2019). This is a classifier that is easy to implement and works well with any type of feature space and as well as implementing for multi-class classification problems but the limitation of KNN is that it is highly computational intensive for large datasets (Chandra Mohan & Baskaran, 2012; Sun et al., 2002). This may implicate that the use of KNN may be suitable for text document classification since it is simple to implement. However, the size of the dataset, such as the number of text feature needs to be taken into consideration as well.

Decision Tree is an algorithm that utilises a hierarchical decomposition technique of the data space where a tree is created based on the attributes of the labelled data points (Aggarwal & Zhai, 2012; De Mántaras, 1991). It is noted that the decision tree algorithm is fast in both learning and prediction but its limitation is the sensitivity of the algorithm to noises and highly susceptible to overfitting as the

decision is constructed solely from the dataset, which may not generalise well to other data (Giovanelli et al., 2017; Quinlan, 1987). As such, this algorithm may not be useful for text document classification that involves wider coverage of topics but great for text document datasets with low noise and highly generalisable data points.

Another tree-based algorithm that is enhanced upon Decision Tree is the Random Forest algorithm. This classifier incorporates ensemble learning method where decision trees generated randomly in parallel where the prediction is then assigned based on voting the best decision tree (Kowsari et al., 2019; Wu et al., 2004). Benefiting from the advantage of decision tree algorithm, the Random Forest classifier is also fast in training for text data sets however is slow in making predictions due to the voting mechanism (Banga et al., 2018).

Finally, the last algorithm is Neural Networks which are deep learning models. Deep learning models are the state-of-the-art results across many domains in classification, including various Natural Language Processing (NLP) applications such as text document classification (Kowsari et al., 2019). Neural Network models are inspired by the biology of neurons, where the model learning is performed through multi-connections of layers.

Every layers between the input and out layers that are called hidden layers receives the output of previous layer as input of one layer (Kowsari et al., 2017). For the application of Neural Network models in text document classification, the input layer or first layer may be fed with vectorised text features such as TF-IDF, word embedding, or other forms of the extracted feature while the output layer consists of the number of classes for in the classification problem, which in the case of text document classification, is the number of topics the documents are classified in.

In the evaluation phase, the confusion matrix is widely used for evaluating the performance of the classification model as exemplified by its usage in multiple studies in their evaluation phases (Arusada et al., 2017; Ikonomakis et al., 2005). According to a systematic analysis by Sokolova & Lapalme (2009), given that this project will require the classification into multiple topics, it is best to evaluate the

overall performance for each class by calculating the macro-average of the same measure for each of the class. The measure that can be used consists of accuracy, precision and recall, which are all derived from the confusion matrix (Sokolova & Lapalme, 2009).

In fact, in a study which reviewed evaluation metrics for classification by M & M.N (2015), a significant limitation of using accuracy as an evaluation metric is that the evaluation score becomes much less distinctive. This limitation is due to the calculation of accuracy which accounts the correct predictions over total instances, which will diminish the importance of a class with lesser number of instances in a data set. Instead of accuracy as an evaluation metric, the study reports the averaged F-Measure to be much more discriminative, as it is calculated based on the average of each class's F-Measure, which weighs down the total score if any single class performs badly. Hence, the use of the averaged F-Measure may be justifiable in this project to ensure a stricter evaluation metric is imposed on the classification algorithms.

2.3 Existing Conference Paper Classification Systems

In the classification system by Mustafa (2020), the classification system uses word2vec model for text representation to determine the semantic and contextual information of words. The system also employed a method to obtain threshold values of the words. The system's prediction uses the similarity threshold where the average similarity score of a test document for every topic is compared with the similarity threshold value.

After that, the topic that is having a score higher than the threshold value or the highest score, is selected as the classified topic to the test document. The classification was performed using two Computer Science research paper datasets from the Journal of Universal Computer Science (JUCS) and Association of Computing Machinery (ACM) respectively. The outcomes revealed good performance in single label classification, where the average accuracy of 0.87 and 0.84 was achieved for the JUCS and ACM datasets, respectively. For multi label classification, the average accuracy is 0.82 & 0.80 for JUCS and ACM, respectively.

In another classification system by Isa et al. (2008), the model implements an enhanced hybrid classification technique by using Naïve Bayes approach and the Support Vector Machine (SVM). The Bayes formula was used to vectorise instead of for classification, where the probability of a document belonging to a category is calculated. The Bayesian formula gives a range of probabilities to which the document can be assigned according to a predetermined set of topics.

These probabilities are the vector feature that are used for the SVM algorithm for classification. According to Isa et al. (2008), the disadvantage of Naïve Bayes classifier that uses only the highest probability can be overcome using the SVM which uses all the probability values associated with every category for each document as a vector feature. The system also resulted in a significant training time reduction compared to classifiers that use distance metrics such as k-Nearest Neighbour and as well as improvement in the classification accuracy when compared to pure Naïve Bayes systems.

Vo & Ock (2015) also implemented a classification system with their proposed framework to classify short text documents based on their titles. Three datasets from Computer Science Bibliography (DBLP), Lecture Notes in Computer Science book series (LNCS), and Wikipedia were used. Topic modelling approach was also used before classification was performed, which is to reduce common topics into a single topic using Latent Dirichlet Allocation (LDA) that uses a generative probability model. After that, two methods were used to enhance features, the first is by assigning topics from topic models as external features and the second by combining external texts of adapted topics as external features. The system uses SVM, Naive Bayes and KNN algorithms for classification. In the results, SVM classifier performed better than Naive Bayes and KNN in classification performance.

All in all, the common traits that is observed in these systems are that conference papers data set must require some form of numerical representation through vectorisation. The trend of SVM outperforming other algorithms can also be

seen in the results of these studies. Therefore, it may be justifiable that the usage of a vectoriser is required in this project and as well as including SVM as one of the algorithms to be evaluated.

2.4 Agile Software Development Methodology

Most modern software development processes nowadays adopt the agile methodology as it offers a lightweight framework for organisations to respond dynamically and iteratively, compared to traditional approaches, which are becoming more obsolete due to its inflexible nature that cannot cater to volatile and unpredictable modern projects (Abd et al., 2016).

In an empirical study of the agile methodology by Dybå & Dingsøyr (2008), the differences in agile and traditional methodologies were clearly defined.

Table 2.4: Differences between traditional and agile approach (Dybå & Dingsøyr, 2008)

	Traditional development	Agile development
Fundamental assumption	Systems are fully specifiable and predictable	Systems are developed with continuous design improvement
Management style	Command and control	Leadership and collaboration
Knowledge management	Explicit	Tacit
Communication	Formal	Informal
Development model	Lifecycle	Evolutionary
Organisational structure	Mechanistic	Organic
Quality control	Late and heavy	Continuous and incremental

Referring to Table 2.4, there are seven categories in which traditional and agile approaches differ. All these categories can define traditional methodology as a rigid, formal and strict approach that assumes systems can be completely defined, predictable and are built through proper planning which is an unrealistic assumption considering the volatility of modern software development (Abd et al., 2016). On the other hand, continuous improvement and feedback-based testing being the core of agile are much more suitable for as a modern software development methodology (James, 2007).

CHAPTER 3
PROJECT METHODOLOGY AND PLANNING

3.1 Introduction

This chapter explains and details the software development methodology, data mining methodology, development tools and project plan of this project.

3.1.1 Agile Software Development Methodology

The software development methodology that is used for this project is one of the Agile software development methodology which is the Feature Driven Development (FDD) methodology.

3.1.2 Feature Driven Development

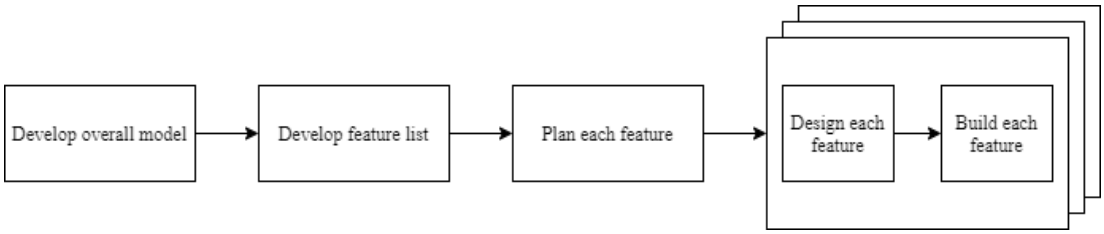


Figure 3.1: Feature Driven Development Process

FDD is a model-driven short-iteration process that consists of five activities. An overall model of the system is designed for the first two sequential activities. Then, the third activity groups the related features together and the last two activities are iterated for each feature.

Table 3.1: Description of each activity in FDD process

Sequential activity	**Description**
Develop an overall model	Create a high-level view of the scope and the architecture of the system
Build a feature list	The overall model is functionally broken down into list of features
Plan by feature	The feature list is then grouped into relevant features called feature sets
Iterative activity	
Design by feature	For each iteration, one feature set is selected, and a detailed design is produced for that feature set.
Build by feature	For the selected feature set, coding and implementation is done. Testing and code inspection are performed once the implementation is done. The completed feature is promoted to the main build of the system once it passes the testing and code inspection.

3.2 Data Mining Methodology

The data mining methodology that is used for this project is one of the data mining process models, which is the Knowledge Discovery in Databases (KDD) process.

3.2.1 Knowledge Discovery in Databases (KDD) process

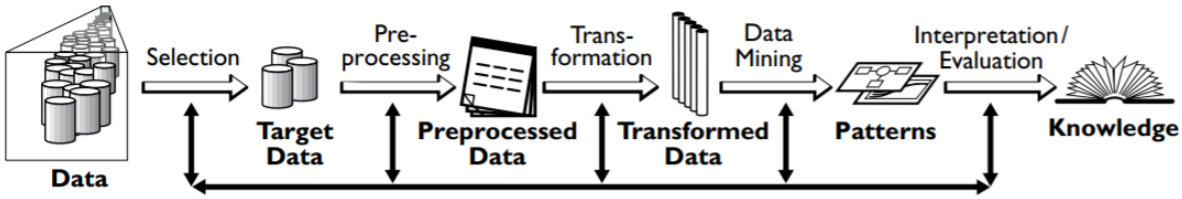


Figure 3.2: KDD process (Fayyad et al., 1996)

The KDD process is similar to how a general data mining workflow should be, as reviewed in Chapter 2. Firstly, the data collected for this project are papers of Artificial Intelligence conference. Five topics of Artificial Intelligence (Natural Language Processing, Unsupervised Learning, Machine Learning, Computer Vision and Pattern Recognition, and Neural and Evolutionary Computing) were picked as the classes for categorising the conference papers. For each topic, 200 conference papers were collected from journal publishers such as Springer, ScienceDirect, IEEEXplore, ACL and ACM (Scopus and ISI Indexed). Papers from these publishers usually are Scopus and ISI indexed which ensure reliability. These conference papers were used for training and testing the data mining models.

In the selection phase of the KDD process, the features to be used for input to the data mining model is selected. For this project, the title and abstract of the conference papers were extracted as the features while the topic of each paper is the class or target feature.

In the pre-processing phase, features were pre-processed to clean and reduce the impurity in the data such as symbols, upper and lower cases that may affect the performance of the data mining model. The text pre-processing techniques that were used are stop words removal, stemming and tokenisation.

In the transformation phase, features were transformed so that they can be the effective input for the data mining model. For text features, they cannot be used as inputs as data mining models require inputs to be numerical. Therefore, text features need to be represented in numbers while trying to maintain the property of the features. In this project, text features were transformed and represented with Term Frequency – Inverse Document Frequency (TF-IDF) which reflects how important a word is to a document compared to the collection of documents.

The next phase will be the data mining phase, where patterns are discovered from the data using data mining algorithms which can then be used to make predictions. The data mining problem in this project is a multi-class classification problem which requires conference papers to be categorised into five classes of topics. For this project, the data mining algorithms that were used are Support Vector Machine, Naïve Bayes, Logistic Regression, K-Nearest Neighbour, Decision Tree and Random Forest. Data mining modelling was then performed using a randomised stratified k-fold validation, where each algorithm was trained in 5 folds of data set where each fold was stratified to ensure equal distribution of classes in each fold.

The final phase of the KDD process will be the evaluation phase, where the performance of the data mining model's prediction is evaluated with certain metrics. In this project, the macro-averaged F-Measure was calculated for each of the data mining algorithms mentioned earlier in the data mining phase. Then, the models that achieved top three in the macro-averaged F-Measure evaluation was then selected for optimisation.

Notice that in Figure 3.1, the flow of the arrow goes back to each of the earlier stages in the KDD process. The flow is to illustrate that the KDD process is iterative, where refinement will be done to the data mining model upon evaluation until a satisfactory result is obtained. The refinement process was performed in this project through grid search optimisation, where a set of parameters for each model is used for each iteration of grid search. Upon the completion of the grid search optimisation, the set of parameters that yields the highest F-Measure score will be returned for the top three algorithms.

Finally, the best performing data mining algorithm among the top three will be chosen and deployed into the web-based system for classifying the conference papers when authors submit their papers. The system then matches reviewer users based on their topic of expertise with the predicted topic of the conference papers. The matched reviewers are then displayed to the chairman users to automate their decision-making process in assigning the conference papers.

3.3 Low-Fidelity Design

A low fidelity design is sketched on the main pages of Login, Chairman user, Reviewer user and Author user. The design is sketched using draw.io, which is a free online diagramming tool. The Login page is attached as Appendix A, Chairman user page as Appendix B, Reviewer user page as Appendix C and Author user page as Appendix D.

3.4 Project Plan

This section outlines the Work Breakdown Structure and Gantt Chart.

3.4.1 Work Breakdown Structure

Work Breakdown Structure is attached as Appendix E.

3.4.2 Gantt Chart

Gantt Chart is attached as Appendix F.

3.5 Development Tools

This section describes the development tools that will be used in the project such as programming language, frameworks, database and integrated development environment (IDE).

3.5.1 Programming Language

Python is an interpreted, high-level, general-purpose programming language. The Python version that is used for this project is the Python 3.5.x version. Python's design philosophy prioritises code readability with the use of whitespaces. The

object-oriented approach of the language will help logical and clear codes to be written in this project.

3.5.2 Frameworks

- **Django**

Django is a server-side web framework, constructed using Python. It is a full-stack and an open-source Python web development framework. Django comes with a variety of ready-to-use libraries that can develop scalable and versatile web applications swiftly.

- **Pandas**

Pandas is a Python library written for data manipulation and analysis. It offers fast, powerful and flexible data structures and operations for manipulating data in tables.

- **NumPy**

NumPy is a Python library that offers comprehensive mathematical functions, random number generators, linear algebra routines and many more for mathematical computations.

- **Scikit-learn**

Scikit-learn is a Python machine learning library. It features many classification, regression and clustering algorithms that are used as data mining tools.

- **NLTK**

The Natural Language Toolkit (NLTK) is a Python library that works with human language data. It comes with a comprehensive set of natural language processing (NLP) tools such as tokenisation, parsing, classification, stemming, tagging and semantic reasoning.

- **Matplotlib**

Matplotlib is a comprehensive Python library for creating static, animated, and interactive visualisations which can be used to report results and performance during evaluation of data mining models.

3.5.3 Database

MySQL database will be used as a database server in this project for SQL operations. MySQL is an open-source relational database management system (RDBMS) with a client-server model. RDBMS is a software or service that will be used to create and manage databases based on a relational model.

3.5.4 Integrated Development Environment

PyCharm is an integrated development environment (IDE) for the Python language. It offers features that can aid in implementing Python applications such code analysis, project navigation, refactoring tools, graphical debugger, testing tools, integration with version control systems and support for web development with Django as well.

CHAPTER 4
PROJECT SPECIFICATION

4.1 Introduction

This chapter is to show the requirements that are derived from studying previous works of the conference paper management systems. Use case diagrams and use case descriptions are also illustrated to show the flow of the web-based system.

4.2 Use Case Diagram

A use case diagram is drawn to illustrate the processes and workflow of the users consisting of authors, chairmen and reviewers.

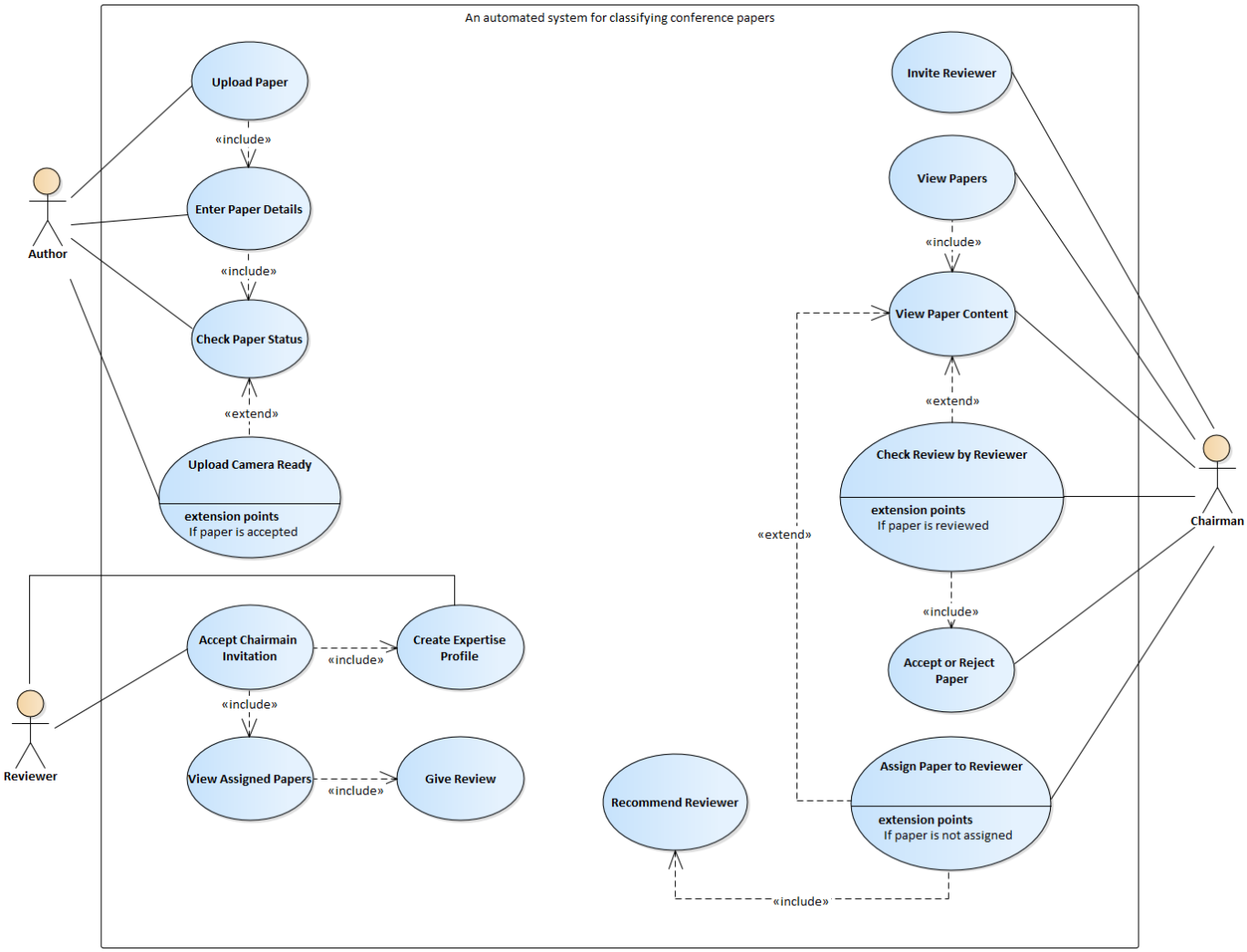


Figure 4.1: Use Case Diagram of the web-based system

4.3 Use Case Descriptions

Table 4.1: Upload Paper use case description

Use Case Name: Upload paper	**ID:** UC01
Primary Actor: Author	
Brief Description: The author uploads paper and enters paper details that will be reviewed by the reviewer and chairman.	
Preconditions: 1. The user should be logged with an author account.	
Relationships: **Association:** Author User **Include:** Enter Paper Details **Extend:** - **Generalisation:** -	
Normal Flow of Events: 1. The author user wants to upload a research paper for a conference. 2. The author uploads the paper in PDF format. 3. The author enters the paper details such as title, keyword and abstract. 4. The author confirms the upload and await to be reviewed.	

Table 4.2: Check Paper Status use case description

Use Case Name: Check Paper Status	**ID:** UC02
Primary Actor: Author	
Brief Description: The author checks the status of the uploaded paper to see if the paper has been accepted or rejected. The author can upload the camera-ready paper once accepted.	
Preconditions: 1. The user should be logged with an author account. 2. The author should have already uploaded the paper for review.	
Relationships: **Association:** Author User **Include:** - **Extend:** Upload Camera Ready **Generalisation:** -	
Normal Flow of Events: 1. The author wants to check the status of the paper. 2. The author checks whether the paper is accepted or rejected along with the reviewer's comment. 3. If paper accepted, the author can proceed to upload the camera-ready paper in PDF format. If rejected, it is the end of process for this upload.	

Table 4.3: Accept Chairman Invitation use case description

Use Case Name: Accept Chairman Invitation	**ID:** UC03
Primary Actor: Reviewer	
Brief Description: The reviewer accepts the chairman's invitation and create expertise profile to start viewing assigned papers to be reviewed by the reviewer.	
Preconditions: 1. The user should be logged in with a reviewer account.	
Relationships: **Association:** Reviewer User **Include:** 1. Create Expertise Profile 2. View Assigned Papers **Extend:** - **Generalisation:** -	
Normal Flow of Events: 1. The reviewer wants to start reviewing papers assigned by chairman. 2. The reviewer accepts a chairman's invitation. 3. The reviewer creates an expertise profile to enter area of expertise. 4. The reviewer views the papers assigned by the chairman to start reading and giving reviews.	

Table 4.4: Give Review use case description

Use Case Name: Give Review	**ID:** UC04
Primary Actor: Reviewer	
Brief Description: After the reviewer has done reading the assigned paper, the reviewer now gives review with the provided options and adds any additional comment.	
Preconditions: 1. The user should be logged in with a reviewer account. 2. The reviewer should have accepted a chairman's invitation. 3. The reviewer should have been assigned with a paper by a chairman.	
Relationships: **Association:** Reviewer User **Include:** - **Extend:** - **Generalisation:** -	
Normal Flow of Events: 1. The reviewer has finished reading the assigned paper and wants to give a review. 2. The reviewer gives a review to the paper with the option of Accept, Weakly Accept, Borderline, Weakly Reject, Reject. 3. If necessary, the reviewer adds additional comments to the reviewed paper.	

Table 4.5: Invite Reviewer use case description

Use Case Name: Invite Reviewer	**ID:** UC05
Primary Actor: Chairman	
Brief Description: The chairman sends invitation to reviewer accounts which they can accept the invitations and start reviewing papers.	
Preconditions: 1. The user should be logged in with a chairman account	
Relationships: **Association:** Chairman User **Include:** - **Extend:** - **Generalisation:** -	
Normal Flow of Events: 1. The chairman wants to invite reviewers to start reviewing papers. 2. The chairman views the list of uninvited reviewer accounts. 3. The chairman sends invitations to the selected reviewer accounts and wait for those reviewers to accept invitation and start assigning papers to them.	

Table 4.6: View Papers use case description

Use Case Name: View Papers	**ID:** UC06
Primary Actor: Chairman	
Brief Description: The chairman views the uploaded papers and then selects a paper to view the content of the paper.	
Preconditions: 1. The user must be logged in with a chairman account.	
Relationships: **Association:** Chairman User **Include:** View Paper Content **Extend:** - **Generalisation:** -	
Normal Flow of Events: 1. The chairman wants to view all the papers and select a paper to view its content. 2. The chairman views the list of the uploaded papers. 3. The chairman selects a paper to view the content of the paper such as the title, keyword, abstract and assigned reviewer if any. 4. The chairman downloads the selected paper for reading. 5. If the paper has been reviewed by a reviewer, the chairman may check the review given and accept or reject the paper. If the paper has not been assigned to any reviewer, the chairman may assign a reviewer.	

Table 4.7: Assign Paper to Reviewer use case description

Use Case Name: Assign Paper to Reviewer	**ID:** UC07
Primary Actor: Chairman	
Brief Description: If the paper is not assigned to any reviewer yet, the chairman will assign a reviewer either manually or automatically as recommended by the system.	
Preconditions: 1. The user must be logged in with a chairman account. 2. The paper does not have any assigned reviewer.	
Relationships: **Association:** Chairman User **Include:** Recommend Reviewer **Extend:** - **Generalisation:** -	
Normal Flow of Events: 1. The chairman wants to assign a reviewer to the paper that has not yet been assigned. 2. If the chairman wants to assign manually, the chairman will select a reviewer from a list of all invited reviewers. If the chairman wants to assign automatically, the chairman will select a reviewer from a list of the recommended reviewers by the system.	

Table 4.8: Recommend Reviewer use case description

Use Case Name: Recommend Reviewer	**ID:** UC08
Primary Actor: System	
Brief Description: If the chairman would like to automatically assign a reviewer, the system will recommend reviewers based on the implemented data mining prediction model.	
Preconditions: 1. The paper does not have any assigned reviewer. 2. The chairman selects the recommended reviewer option.	
Relationships: **Association:** System **Include:** - **Extend:** - **Generalisation:** -	
Normal Flow of Events: 1. The chairman wants to assign the reviewer to the paper automatically. 2. The chairman selects the option to recommend reviewer. 3. The system performs prediction to find suitable reviewers based on the implemented data mining prediction model to match the reviewer's area of expertise and paper's topic. 4. The system generates a list of recommended reviewers for the chairman to select.	

Table 4.9: Check Review by Reviewer use case description

Use Case Name: Check Review by Reviewer	**ID:** UC09
Primary Actor: Chairman	
Brief Description: After the chairman has viewed the paper content, the chairman checks the review on the paper if the paper has been reviewed by its assigned reviewer.	
Preconditions: 1. The user must be logged in with a chairman account. 2. The selected paper should have been reviewed by a reviewer	
Relationships: **Association:** Chairman User **Include:** Accept or Reject Paper **Extend:** - **Generalisation:** -	
Normal Flow of Events: 1. The chairman wants to check the review of the selected paper. 2. The chairman views the review option assigned by the reviewer which is either Accept, Weakly Accept, Borderline, Weakly Reject or Reject. 3. If any additional comment is given by the reviewer, the chairman views the additional comment to further understand the review. 4. The chairman decides to Accept or Reject the selected paper for the conference.	

4.4 Software Requirements

This section specifies the functional and non-functional requirements of the automated system for classifying conference papers.

4.4.1 Functional Requirements

- **General**

1. The system shall allow users to sign up for a Chairman, Reviewer or Author account.
2. The system shall allow users to login to their own account using username and password.

- **Chairman**

1. The system shall allow chairman users to invite reviewers to review papers.
    a. The system shall display the list of uninvited reviewers to the chairman user.
    b. The system shall allow the chairman user to select the reviewers to be invited.
    c. The system shall allow the chairman users to invite the selected users.
2. The system shall allow chairman users to view the uploaded papers.
    a. The system shall display the list of uploaded papers to the chairman user.
    b. The system shall allow the chairman user to filter the list based on assigned, unassigned and reviewed categories.
3. The system shall allow chairman users to assign reviewers to unassigned papers.
    a. The system shall allow the chairman user to manually select a reviewer from a list of all the invited reviewers.
    b. The system shall recommend a list of reviewers for the chairman user to automatically select a reviewer.
4. The system shall allow chairman users to view the paper content.
    a. The system shall allow the chairman user to select a paper to view its content.

   b. The system shall display the content of the paper such as title, keyword, abstract and download link.
   c. The system shall allow the chairman user to download the PDF file of the paper.
5. The system shall allow chairman users to view the review of the paper.
   a. The system shall display the review option given by the assigned reviewer that is Accept, Weakly Accept, Borderline, Weakly Reject or Reject.
   b. The system shall display the additional comments given by the assigned reviewer.
6. The system shall allow the chairman user to select Accept or Reject option for the status of the reviewed paper.

- **Reviewer**

1. The system shall allow reviewer users to accept an invitation from chairman users.
   a. The system shall allow the reviewer user to navigate to the invitation tab to accept invitations from chairman users.
2. The system shall allow reviewer users to create an expertise profile by inputting the area of expertise.
3. The system shall allow reviewer users to view assigned papers.
   a. The system shall display the list of papers assigned to the reviewer user.
   b. The system shall allow the reviewer user to select a paper from the list to view the content such as title, keyword, abstract and download link.
   c. The system shall allow the reviewer user to download the PDF file of the paper.
4. The system shall allow reviewer users to give a review to the assigned paper.
   a. The system shall allow the reviewer user to give one of the review options consisting of Accept, Weakly Accept, Borderline, Weakly Reject and Reject.
   b. The system shall allow the reviewer user to provide additional comments in text form for the paper.

- **Author**

1. The system shall allow author users to upload their paper to be reviewed.
    a. The system shall allow the author user to upload their paper as a PDF file.
    b. The system shall allow the author user to enter details of the paper such as title, keyword and abstract.
2. The system shall allow author users to check the status of the uploaded papers.
    a. The system shall display a list of the uploaded papers by the author user.
    b. The system shall allow the author user to select one of the papers to view the status of the paper review.
    c. The system shall display whether the paper is Accepted or Rejected, and any additional comments given by the assigned reviewer.
        i. The system shall allow the author user to upload the camera-ready paper for the papers with Accepted status.

4.4.2 Non-functional Requirements

1. The system shall prevent unauthorised login attempts on any user's account.
2. The system shall be fast and responsive.
    a. The navigation user interface (UI) of the system shall not have any delay exceeding 3 seconds.
3. The system shall be able to operate on a web browser.
4. The system shall be able to interoperate between different web browsers.

CHAPTER 5
SYSTEM DESIGN

5.1 Introduction

This chapter describes the system architecture design, database design and as well as the user interface (UI) of the system.

5.2 System Architecture Design

Previously, the system architecture was described in chapter one as a concept. In this section, the implementation for each of the layer will be discussed.

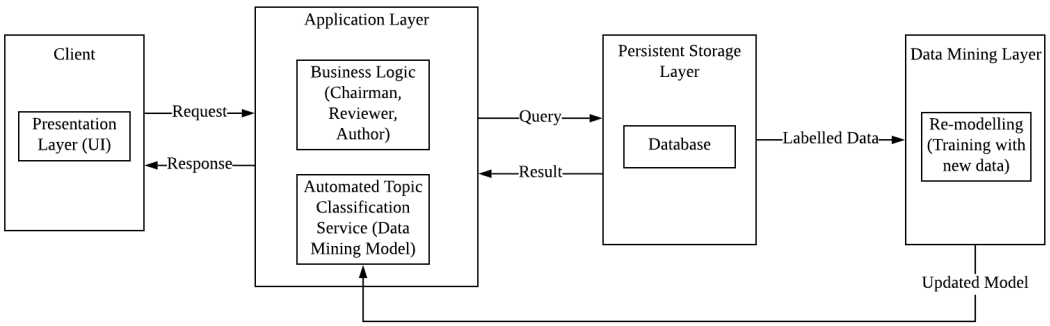


Figure 5.1: An Enhanced 3-Tier Web-based System with Data Mining Layer

Referring to Figure 5.1, the project was developed upon this architecture with four layers, which are the client, application, persistent storage and data mining layer. The data mining layer is an enhancement upon the traditional 3-Tier architecture. This architecture is implemented with Django, which is a Python-based web framework based on a Model-View-Template (MVT) design. The MVT design of Django offers many built-in classes that reduces the complexity in implementing our system architecture, such as eliminating the need to write low-level codes for handling HTTP requests, constructing raw SQL queries and defining the connection with the database.

5.2.1 Presentation Layer

The presentation layer is the client-facing layer. This layer is where the web pages are rendered at the client's side through a web browser. In this project, each web page is created using the Django web framework's templates. Pages are built upon Django templates, which are HTML files that can be embedded with Python codes. The ability to embed Python codes into the HTML files allow conditions, looping and object data to be used in the templates for rendering UIs and data based on the application logics (chairman, reviewer and author accounts).

5.2.2 Application and Persistent Storage Layers

For the application layer, it resides with the web server. This layer is where the business logic such as processes of author, reviewer and chairman accounts are performed upon the requests from the client. As for the persistent storage layer, it resides with the MySQL database server. The implementation of the application layer is through Django's ability to define pages as View objects.

Each View object contains predefined methods to handle different types of HTTP requests such as GET and POST requests. Each View object can also be associated with a self-defined object model such as paper, conference, review, user, chairman, reviewer and author objects. This association allows data to be queried automatically from and to the database when performing actions in the View objects such as creating, retrieving, updating and deleting any of the object models.

This is the main benefit of using the Django framework, where there a layer of abstraction on top of object models to directly associate themselves to the database tables to make queries without constructing any raw SQL statements. This abstraction by the Django framework helps to make interaction between the application and persistent storage layer much closer and simpler. It is just a matter of calling an object's method to get certain fields of the object model from the database tables or to save the user inputs as an insert or update into the database tables.

In this layer, one of the View objects that is associated with the paper's object model is where the prediction of the topic takes place. It is called the Automated Classification Service in this system architecture. This was achieved by exporting the both the vectoriser and the fine-tuned SVM model as a file, which is then loaded into the web application through the Django's configuration file. It is also one of the main reasons a Python framework was chosen to develop the web application, as the data mining model can be loaded natively, since both the model and web application are written in the same language. Hence, the complexity of applying the data mining model's prediction as an external API can be prevented.

5.2.3 Data Mining Layer

In the data mining layer, the data mining model is created through the KDD process. The initial output of the data mining layer is a vectoriser and a newly trained data mining model which are exported as a pickle file using the pickle library that allows a Python object to be serialised into a file, and then loading back the Python object by de-serialising the file back into an object in the web application.

There are two Python objects that are being exported as a pickle file. The first object is the TF-IDF vectoriser in order to vectorise the title and abstract of submitted papers into numerical features. The second object is the SVM classifier as the data mining model, to predict the submitted paper's topic. The pickle file is then put into a directory of the web application and the file is de-serialised into two variables, where one is the vectoriser and another is the SVM classifier. They are then available readily as objects to be called in the application layer for performing topic predictions.

The process of re-training in the data mining layer is also possible. Consider the scenario where new conference papers are being accumulated from the submissions of authors. These papers are also labelled since their topics have been predicted by the system and corrected by the chairmen if any predictions were wrong. Therefore, this new archive conference papers in the database can be queried to the data mining layer to re-train the model with new data through the KDD process again. This re-training process ensures that the model can continuously be improved and

also adjust to any changes in the pattern and correlations between the features and labels.

5.3 Database Design

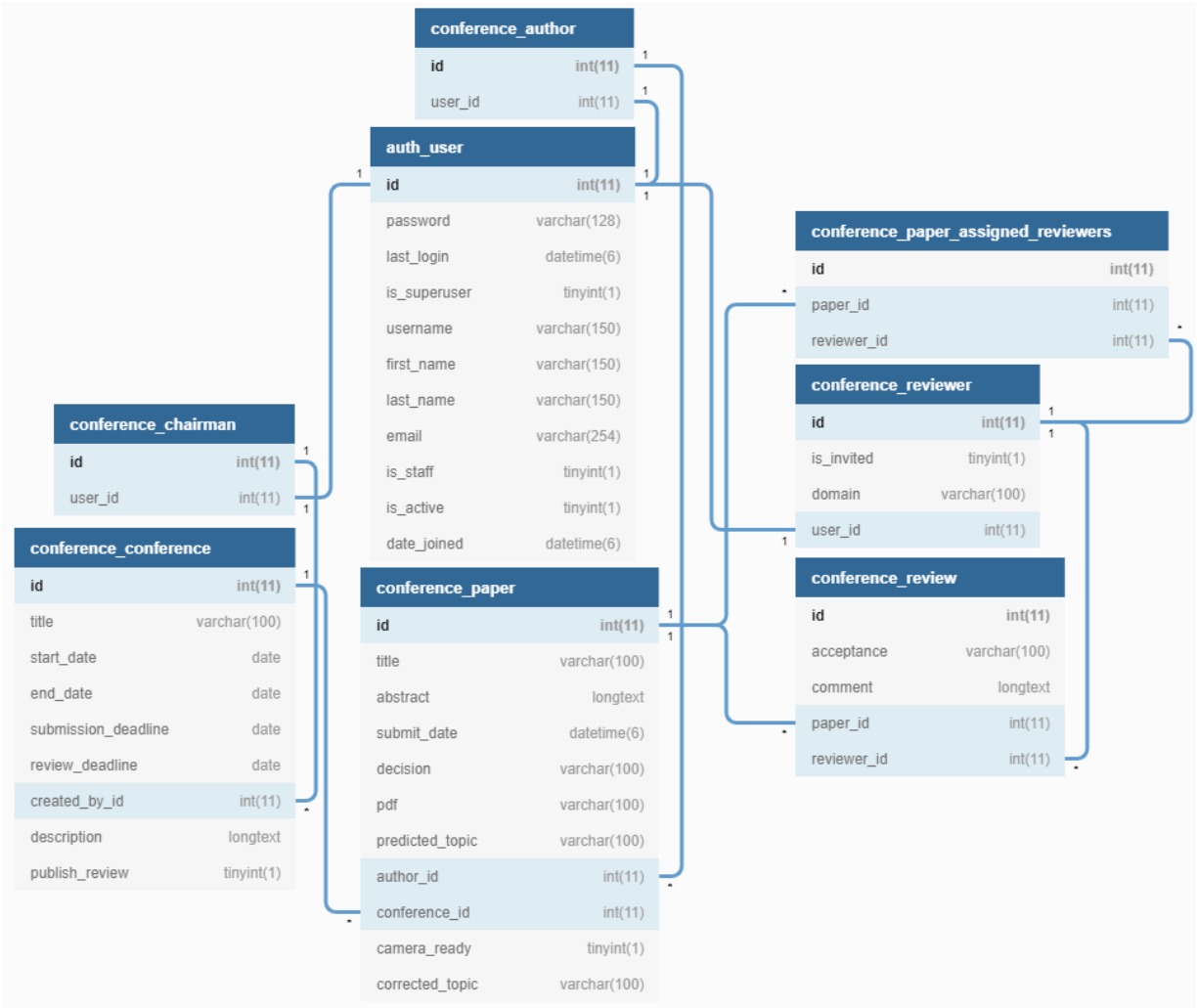


Figure 5.2: Entity Relationship Diagram

5.4 Web Application User Interface

5.4.1 Common views

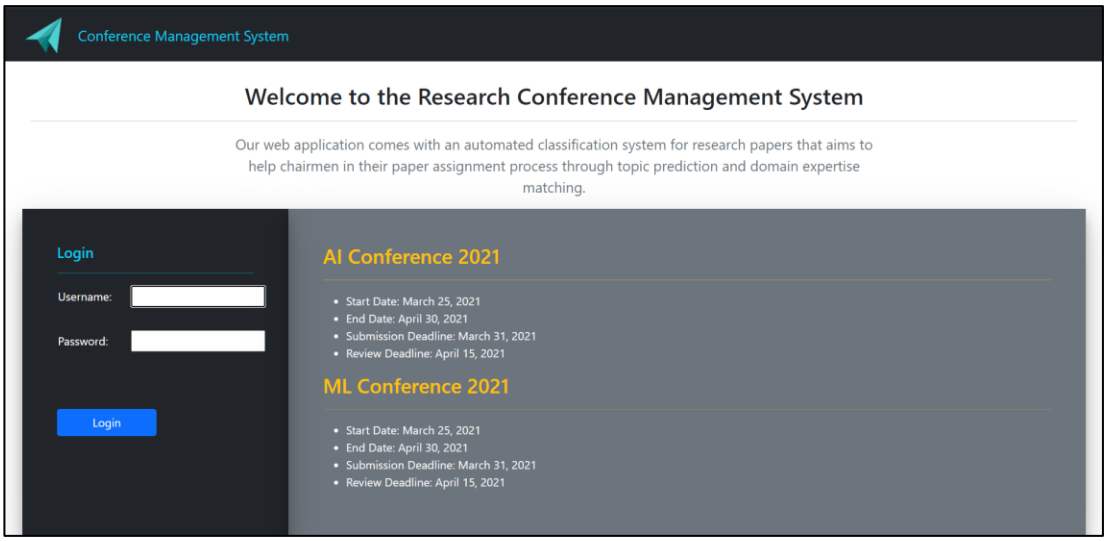


Figure 5.3: Login page

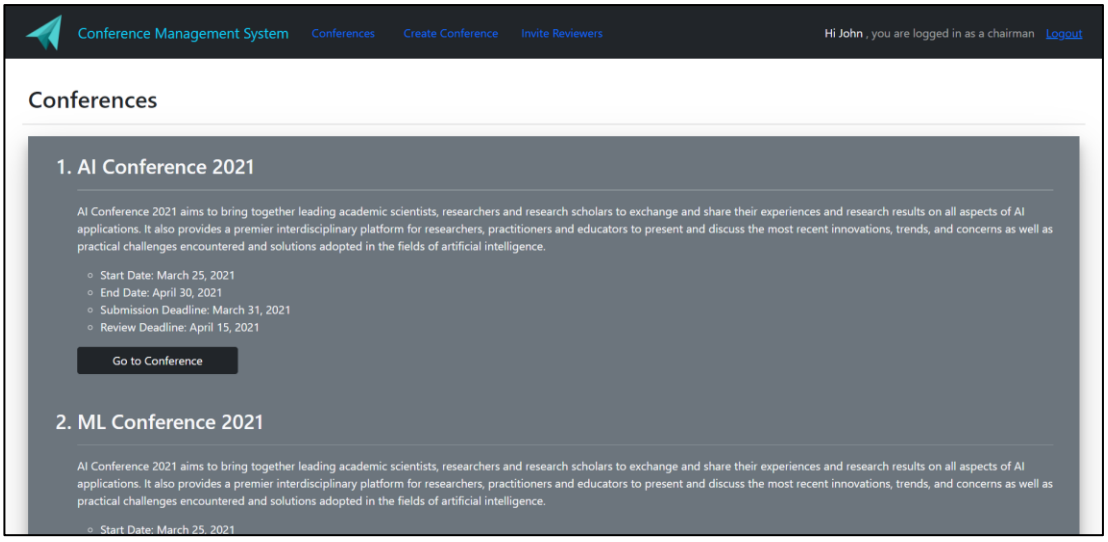


Figure 5.4: Conferences

5.4.2 Chairman's views

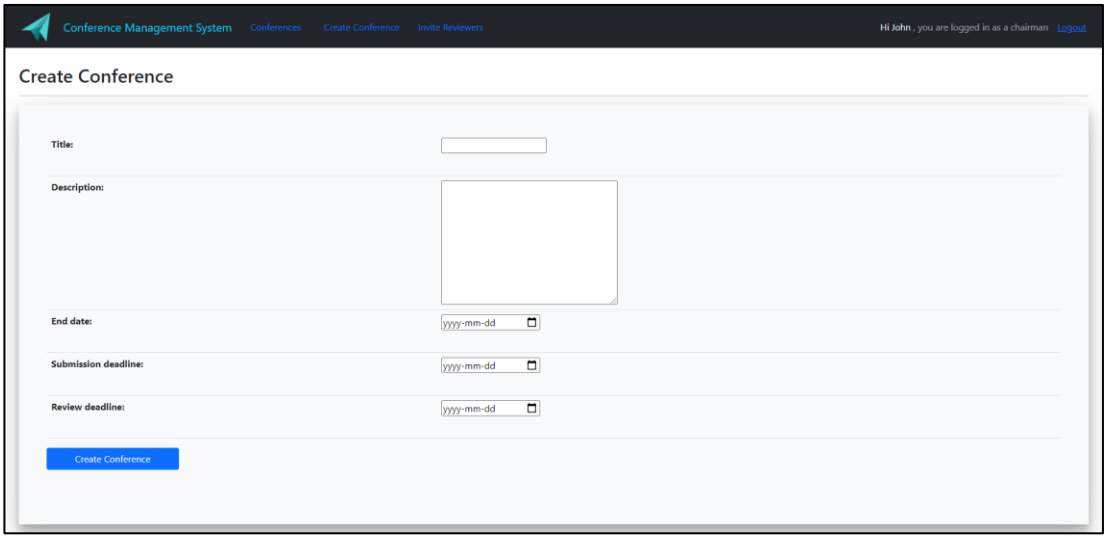


Figure 5.5: Conference creation

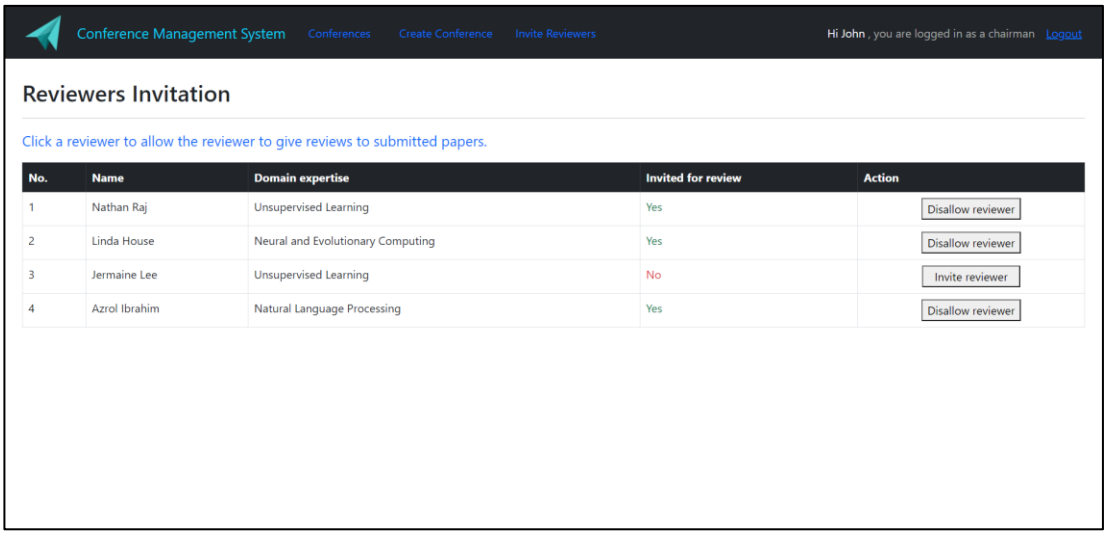


Figure 5.6: Reviewers invitation

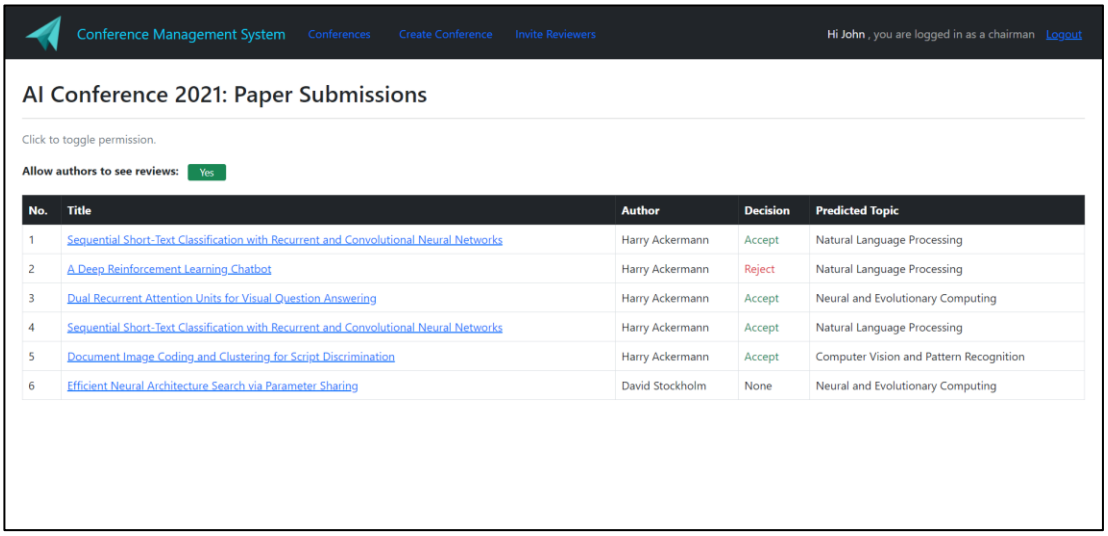


Figure 5.7: Paper submissions under a conference

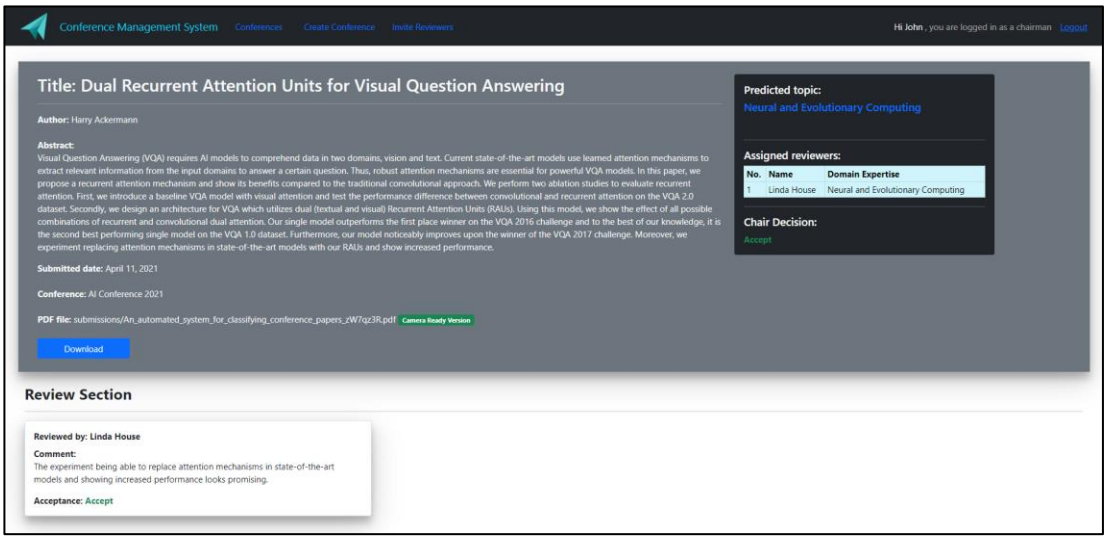


Figure 5.8: Paper details and reviews

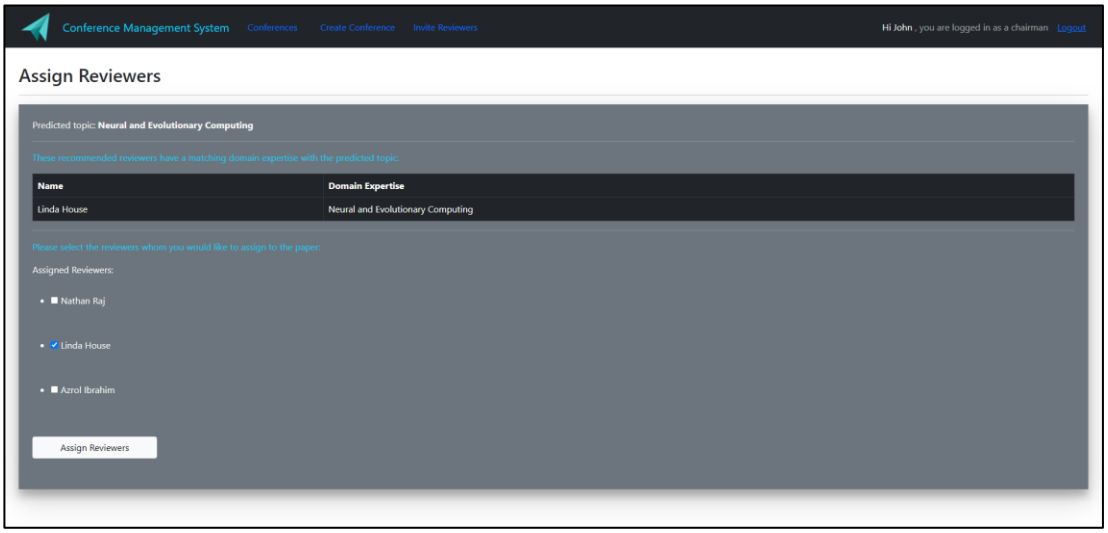


Figure 5.9: Paper assignment

5.4.3 Author's views

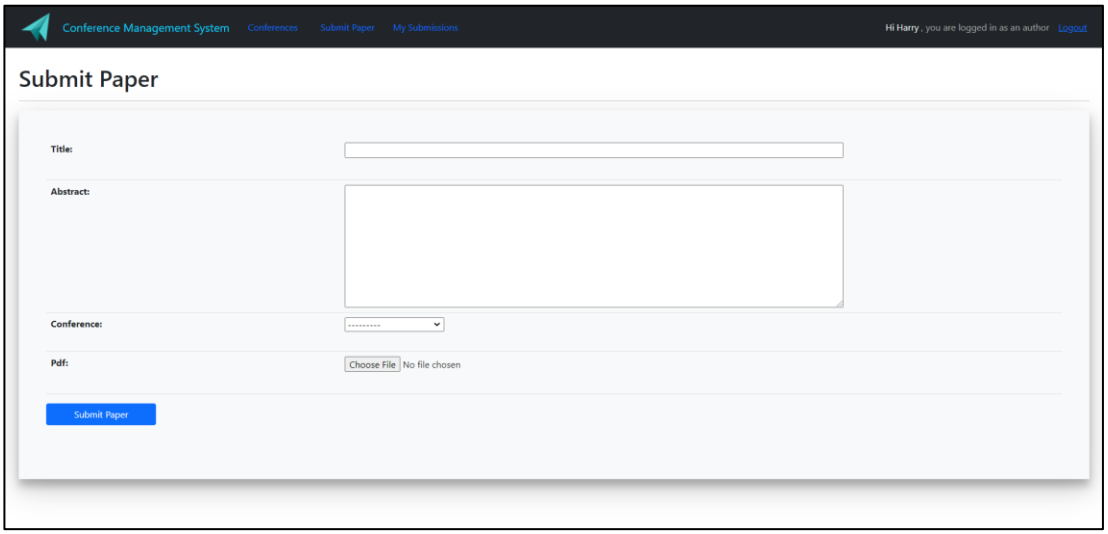


Figure 5.10: Paper submission

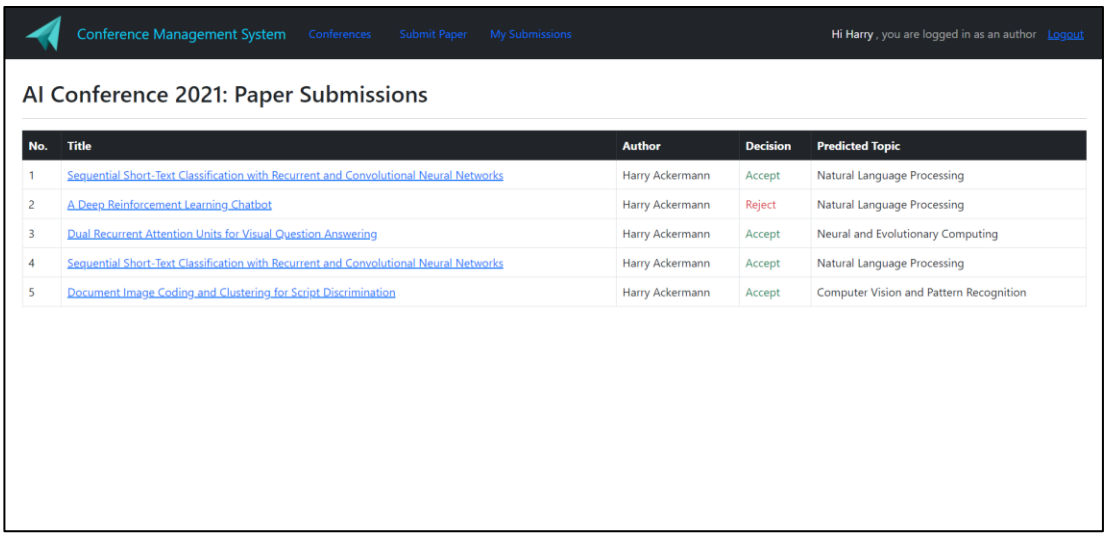


Figure 5.11: Submissions of an author under a conference

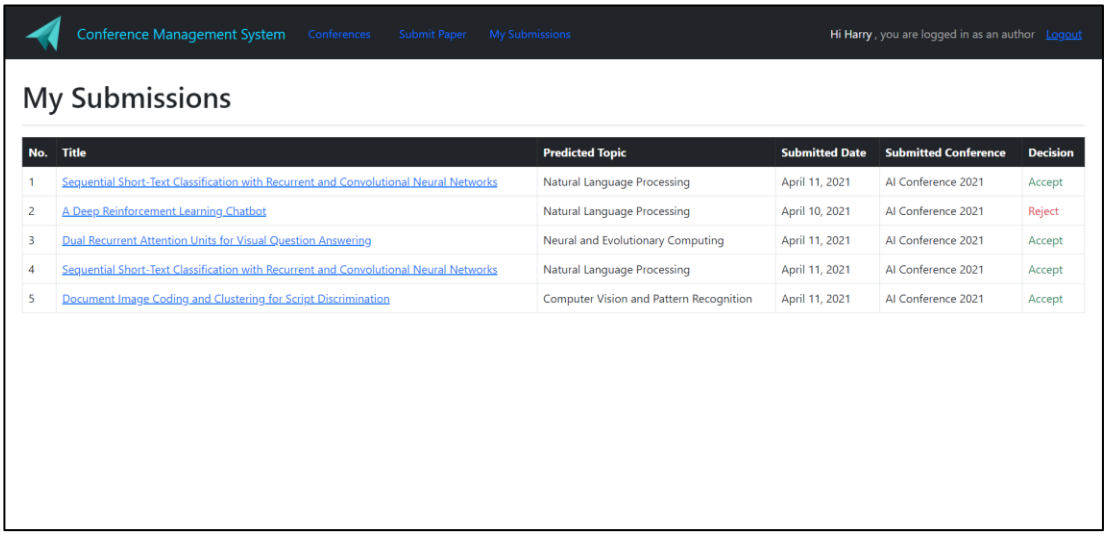


Figure 5.12: All submissions of an author

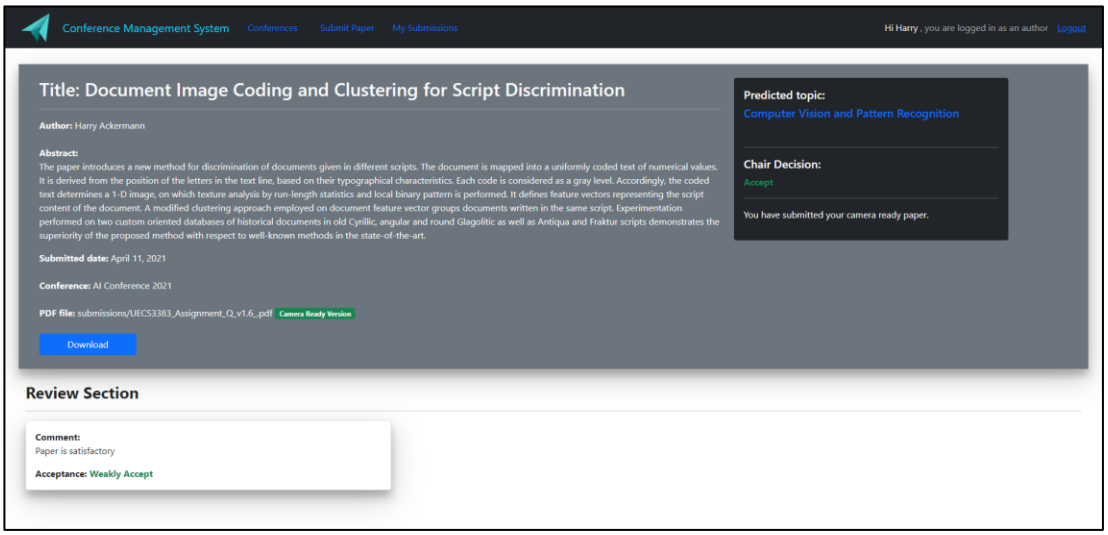


Figure 5.13: Paper details and reviews

5.4.4 Reviewer's views

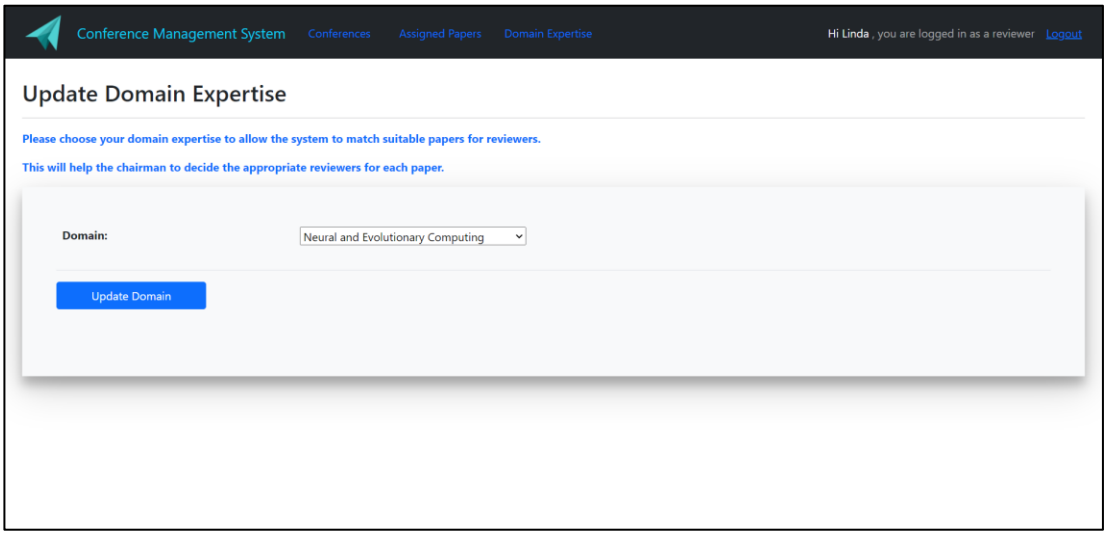


Figure 5.14: Update domain expertise

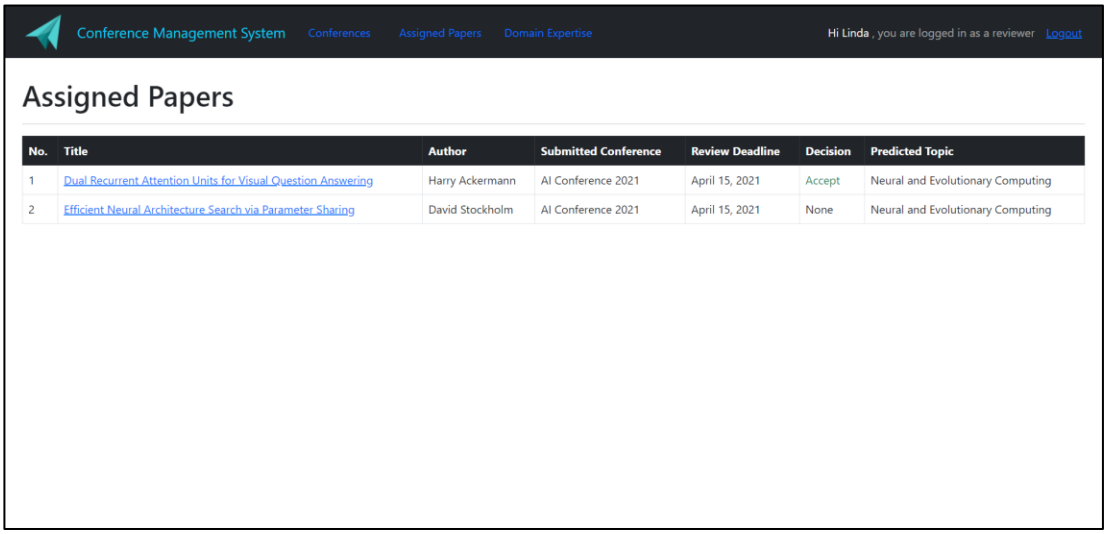


Figure 5.15: Assigned papers

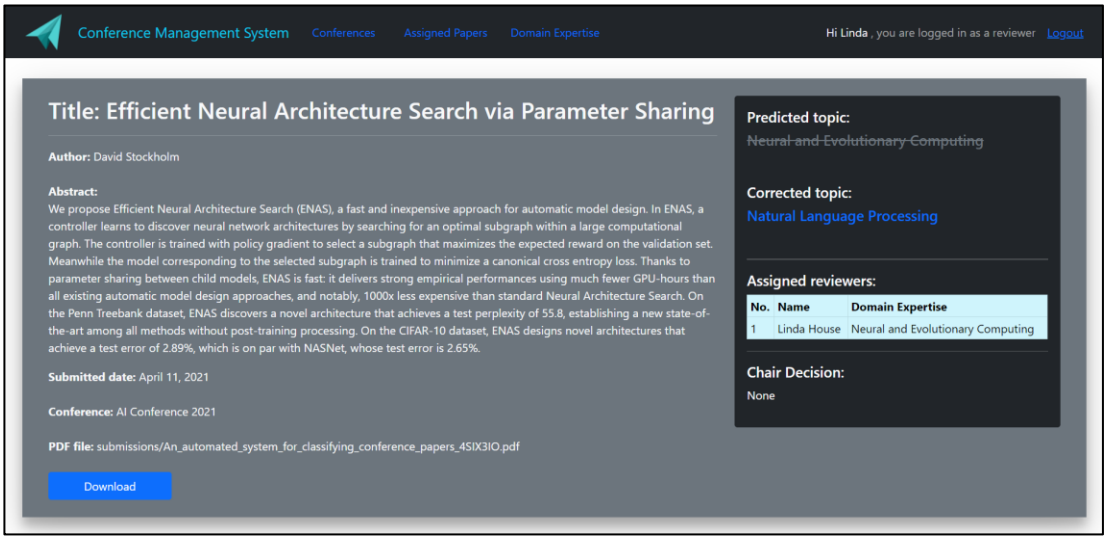


Figure 5.16: Paper details and reviews

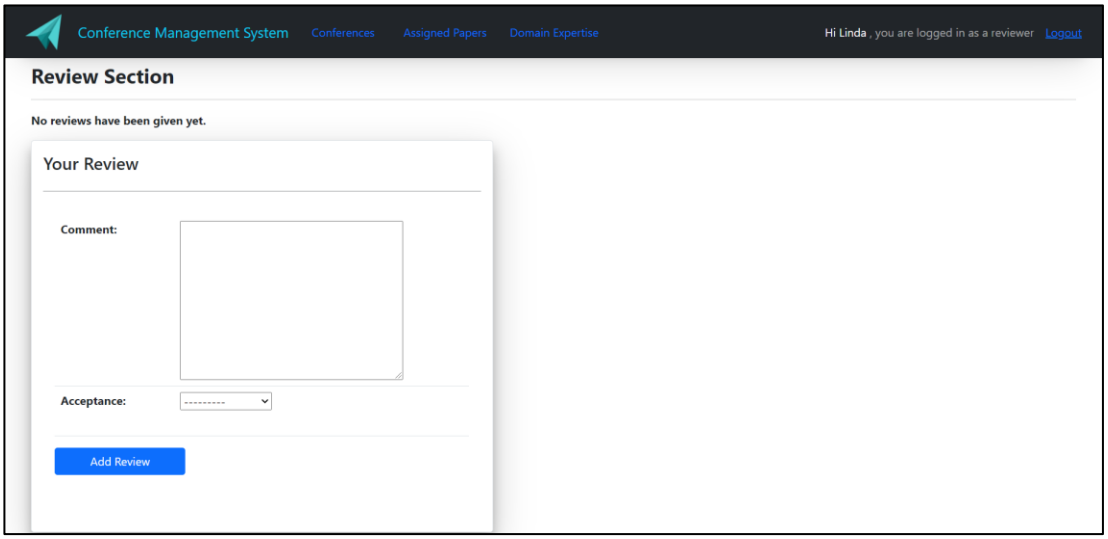


Figure 5.17: Giving a review

CHAPTER 6

PROJECT IMPLEMENTATION

6.1 KDD Process Implementation

In this section, the implementation of each step of the KDD process is described, which consists of selection, pre-processing, transformation, data mining and evaluation.

6.1.1 Selection

The first step of the KDD process is to perform selection. As mentioned before in Chapter 3 for the data mining methodology, the data that is collected are Artificial Intelligence conference papers. In this project, the papers were selected through an intricate filtering process to avoid inaccurate labels that would affect the accuracy of the data mining model. The five AI topics () that were chosen as classes for categorising the conference papers was searched on Arxiv, which is an open-source journal database by the Cornell University.

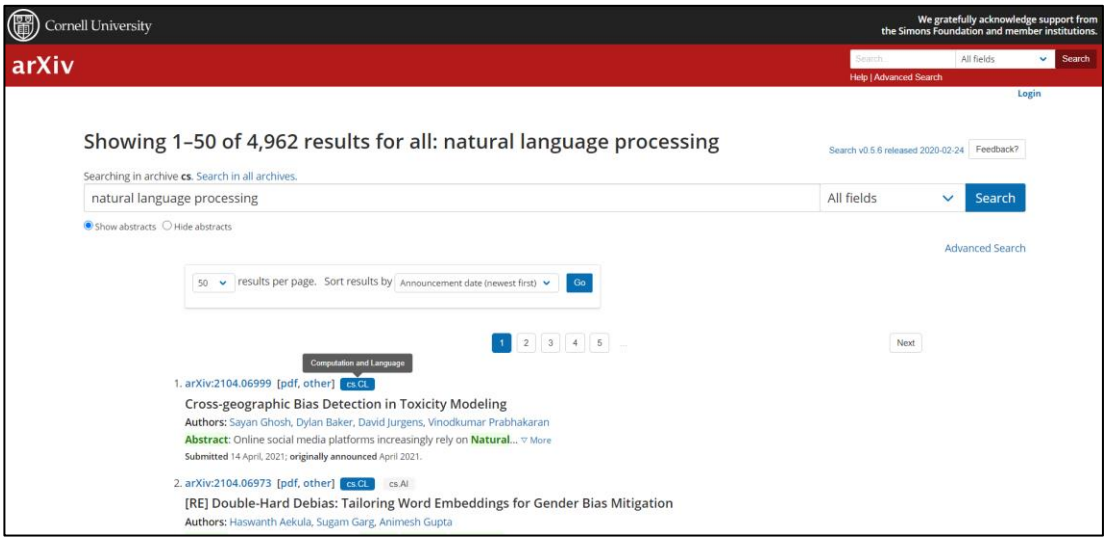


Figure 6.1: Arxiv open-source journals by Cornell University

For each of the papers taken from Arxiv, its title is then searched on Google, to find out whether it is published to any of the journal publishers (Springer, ScienceDirect, IEEEXplore and ACM). If the paper does not exist in any of the publishers' archive, it is skipped. If the paper exists in the publishers' archive, the paper is then checked that they are published under a conference proceeding. This

filtration process ensures the reliability of the labels for our data set. For the papers that pass this filtration process, their titles and abstracts are taken as features, while their topic is taken as the label which is one of the five classes (Natural Language Processing, Unsupervised Learning, Machine Learning, Computer Vision and Pattern Recognition, and Neural and Evolutionary Computing).

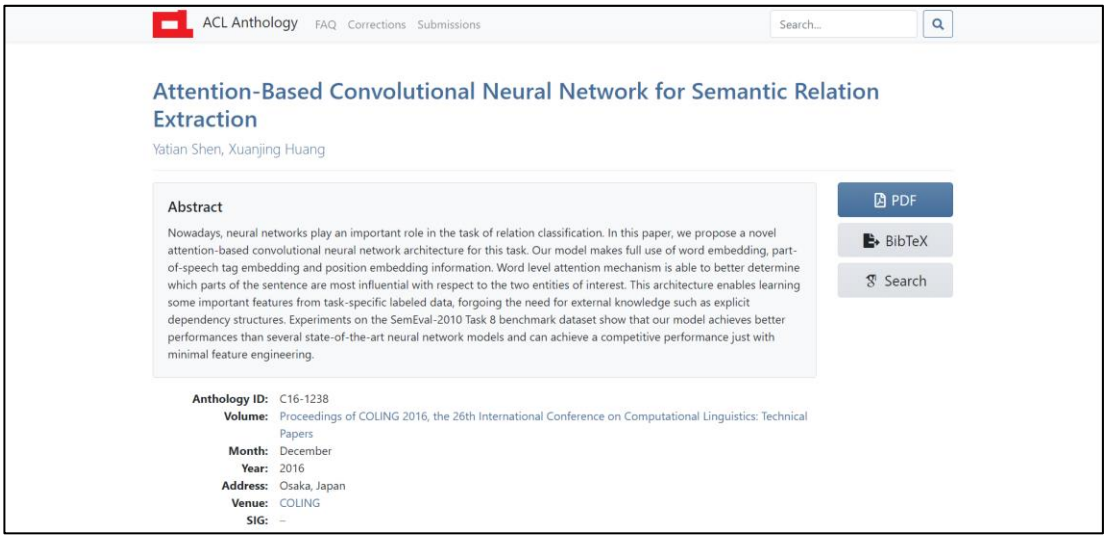


Figure 6.2: Check paper for conference proceedings under a publisher

At the end of the selection process, 200 conference papers for each class were collected.

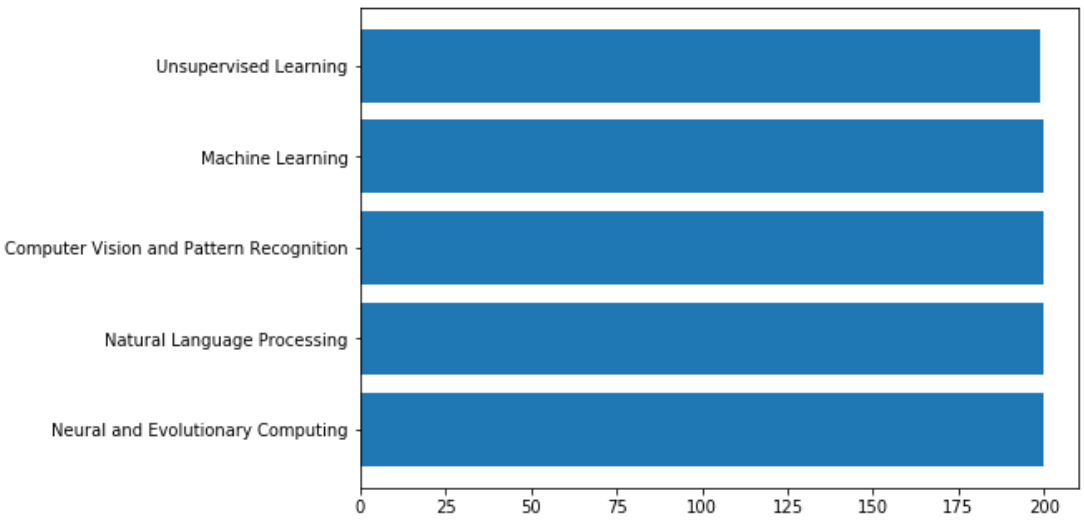


Figure 6.3: Conference papers distribution among classes

6.1.2 Pre-processing

In the pre-processing phase, text pre-processing techniques were used to remove symbols and reduce the variations in the characters that causes duplicate words such as upper and lower cases which do not contribute meaningful patterns for the data mining algorithm. The text pre-processing techniques that were used are symbols removal, stop words removal and tokenisation.

```
#remove non-alphabetic characters
dataset['Abstract'] = dataset['Abstract'].str.replace('[^a-zA-Z]', ' ')
```

Figure 6.4: Symbols removal

```
#declare tf-idf vectoriser
#include stop word removal
#include lower casing
tfidf = TfidfVectorizer(ngram_range=(1,1),
                        stop_words='english',
                        lowercase=True,
                       )
```

Figure 6.5: Stop words removal and lower casing

The stop words removal and lower casing are included in a TF-IDF vectoriser instance itself, so it is defined together with the declaration of a vectoriser.

6.1.3 Transformation

In the transformation phase, features were transformed into numerical representations of each word using the TF-IDF vectoriser. TF-IDF measures how relevant a word is to a paper compared to other papers. It is calculated by multiplying Term Frequency (How many times a word appears in a paper), and the Inverse Document Frequency (Inverse of how many times a word appears across a set of papers). For each word, the TF-IDF ranges from 0 to 1. The higher the TF-IDF value is, the more significant the word is towards that paper compared to the rest of the papers.

```
#declare tf-idf vectoriser
#include stop word removal
#include lower casing
tfidf = TfidfVectorizer(ngram_range=(1,1),
                        stop_words='english',
                        lowercase=True,
                       )

#apply transformation to dataset
features = tfidf.fit_transform((dataset['Abstract'] + dataset['Title']).values.astype('U')).toarray()
```

Figure 6.6: Applying TF-IDF vectorisation

```
recurrent 0.25290512288568273
propose 0.023937206110856587
paper 0.02157288646267497
powerful 0.04895620883430988
essential 0.06185066928178487
robust 0.04296062616228937
certain 0.054142019158321474
answer 0.05890098085323417
input 0.03683166655833159
information 0.03200658049863101
relevant 0.04788846909754403
extract 0.058266391514151084
mechanisms 0.1696371003905575
attention 0.4230840039057833
```

Figure 6.7: Sample TF-IDF values of a paper

6.1.4 Data Mining

The next phase of the KDD process is the data mining phase, where patterns are discovered from the conference papers using data mining algorithms. The task in this phase is a multi-class classification problem where conference papers are categorised into five classes of topics. The TF-IDF values that was obtained from earlier phases are used as the input data for the data mining algorithms.

The algorithms that were applied on the features are Support Vector Machine, Naïve Bayes, Logistic Regression, K-Nearest Neighbour, Decision Tree and Random Forest which are imported from the Scikit-learn library. For each of the algorithm, a randomised stratified k-fold validation is applied for performing validation. With the randomised stratified k-fold validation, each algorithm was trained in 5 folds of data set where each fold was stratified to ensure equal distribution of classes for each of the folds.

```
folds = StratifiedKFold(n_splits=5, shuffle=True, random_state=5)

models = [
    SVC(kernel='rbf'),
    MultinomialNB(),
    LogisticRegression(),
    KNeighborsClassifier(),
    DecisionTreeClassifier(),
    RandomForestClassifier()
]

results = []
for model in models:
    model_name = model.__class__.__name__
    score = cross_val_score(model, features, labels, cv=folds, scoring='f1_macro').mean()
    results.append([model_name, score])
```

Figure 6.8: Cross-validation of each algorithm

The evaluation metric that was used for validating the algorithms is the macro-averaged F-measure. It is calculated the by averaging the precision and recall values of each class, where the precision is the number of true positive instances divided by the number of predicted positive instances, and the recall is the number of true positive instances, divided by the number of all instances that should have been predicted as positive. The implementation of this metric can be used by specifying the 'scoring' parameter of the cross-validation method as shown in Figure 6.8. All the algorithms were cross-validated, and the results were evaluated in the next phase.

6.1.5 Evaluation and Results

In this phase, the performance of the data mining model's prediction is evaluated with the macro-averaged F-measure. For all the algorithms, the default parameters are first used. Then, the models that achieved top three in the macro-averaged F-measure evaluation was then selected to be further optimised in the next phase to identify the best performing data mining algorithm.

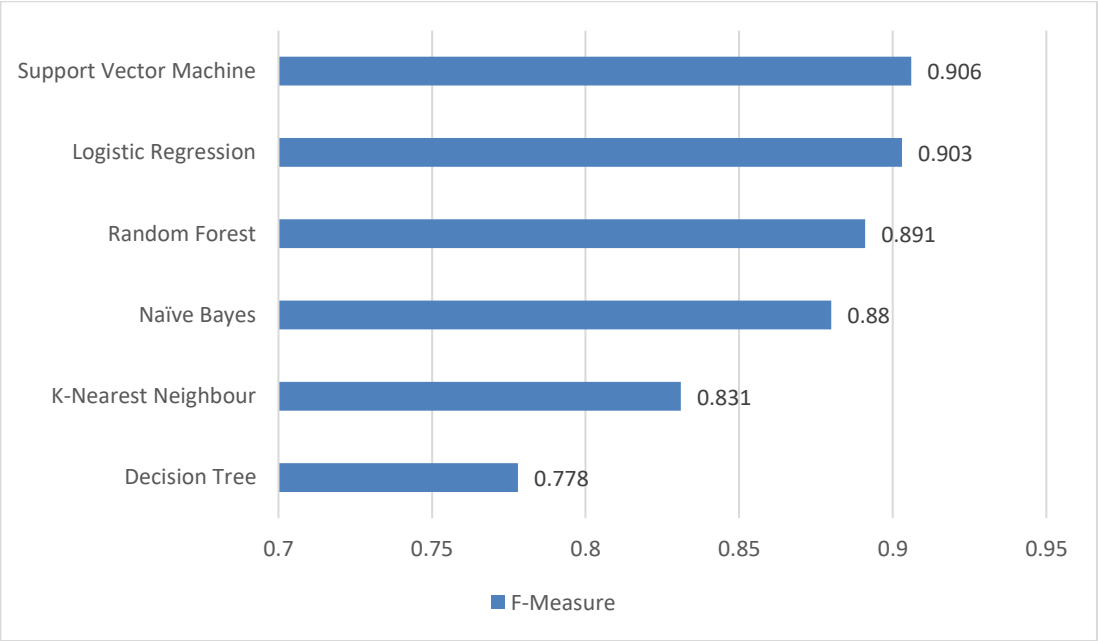


Figure 6.9: F-Measure of each data mining algorithm

Referring to the results of the evaluation in Figure 6.9, SVM outperformed all the other algorithms, standing at 0.906. The F-measure is followed closely by Logistic Regression (0.903), Random Forest (0.891), Naïve Bayes (0.880), K-Nearest Neighbour (0.831) and lastly, Decision Tree (0.778). All the algorithms recorded relatively high F-measures, which indicate that the usage of TF-IDF vectorisation as the numerical representation of the data set is highly effective. However, the Decision Tree was an exception, which recorded a subpar F-measure. The result may be due to the rule-based nature of the algorithm, which could not cater to the complexity of the data set.

Finally, the three best performing algorithms (Support Vector Machine, Logistic Regression and Random Forest) were selected to be further optimised and the results are then again evaluated, to determine the best performing models and its set of optimised parameters.

6.1.6 Optimisation and Results

In this project, the optimisation process was performed through grid search optimisation imported from Scikit-learn. In the grid search optimisation, a set of parameters is defined for each data mining algorithm. For each iteration of the cross-validation, a combination of the parameters is used in the algorithm's instance. The use of different combinations of parameters is repeated until all the combinations of the specified parameters are exhausted.

For the SVM model, the following set of parameters were defined:

- Kernel: 'rbf'
- Gamma: [0.00000001, 0.000001, 0.0001, 0.01, 1, 100, 10000, 1000000, 100000000]
- C: [0.00000001, 0.000001, 0.0001, 0.01, 1, 100, 10000, 1000000, 100000000]

For the Elastic-Net LR model, the following set of parameters were defined:

- Penalty: 'elasticnet'
- C: [0.000001, 0.0001, 0.01, 1, 100]

- Solver: ‘saga’
- L1-ratio: [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]
- Max iteration: [10000]

For the Lasso and Ridge LR model, the following set of parameters were defined:

- Penalty: [‘l1’, ‘l2’]
- C: [0.00000001, 0.000001, 0.0001, 0.01, 1, 100, 10000, 1000000, 100000000]
- Solver: ‘saga’

As for the RF model, the following set of parameters were defined:

- Max depth: [10, 20, 30, 40, 50]
- Max features: ['auto', 'sqrt']
- Minimum samples for a leaf: [1, 4]
- Minimum samples for a split: [2, 5, 10]
- Number of estimators: [200, 400, 600, 800, 1000]

Figure 6.10 below shows the implementation of the grid search for the SVM model. Since the three best performing models are selected for further optimisation, the same grid searching process is repeated for the other two algorithms as well, as shown in Figure 6.11, 6.12 and 6.13. For Logistic Regression, two separate grid search instances were used as some of the parameters of the Elastic-Net Regression do not exist in the Lasso and Ridge Regressions.

```
# grid searching and cross validate using f1_macro scoring
svm_parameters = {
    'kernel':['rbf'],
    'gamma':[0.00000001, 0.000001, 0.0001, 0.01, 1, 100, 10000, 1000000, 100000000],
    'C':[0.00000001, 0.000001, 0.0001, 0.01, 1, 100, 10000, 1000000, 100000000]
}

folds = StratifiedKFold(n_splits=5, shuffle=True, random_state=5)

svm = SVC()
gridsearch_svm = GridSearchCV(svm, svm_parameters, cv=folds, verbose=2, scoring='f1_macro').fit(features, labels)
```

```
Fitting 5 folds for each of 81 candidates, totalling 405 fits
[CV] END ..................C=1e-08, gamma=1e-08, kernel=rbf; total time=   5.7s
[CV] END ..................C=1e-08, gamma=1e-08, kernel=rbf; total time=   5.5s
[CV] END ..................C=1e-08, gamma=1e-08, kernel=rbf; total time=   5.6s
[CV] END ..................C=1e-08, gamma=1e-08, kernel=rbf; total time=   5.6s
[CV] END ..................C=1e-08, gamma=1e-08, kernel=rbf; total time=   5.5s
[CV] END ..................C=1e-08, gamma=1e-06, kernel=rbf; total time=   5.3s
[CV] END ..................C=1e-08, gamma=1e-06, kernel=rbf; total time=   5.5s
[CV] END ..................C=1e-08, gamma=1e-06, kernel=rbf; total time=   5.4s
[CV] END ..................C=1e-08, gamma=1e-06, kernel=rbf; total time=   5.4s
[CV] END ..................C=1e-08, gamma=1e-06, kernel=rbf; total time=   5.4s
[CV] END .................C=1e-08, gamma=0.0001, kernel=rbf; total time=   5.3s
[CV] END .................C=1e-08, gamma=0.0001, kernel=rbf; total time=   5.4s
[CV] END .................C=1e-08, gamma=0.0001, kernel=rbf; total time=   5.5s
[CV] END .................C=1e-08, gamma=0.0001, kernel=rbf; total time=   5.4s
[CV] END .................C=1e-08, gamma=0.0001, kernel=rbf; total time=   5.6s
[CV] END ...................C=1e-08, gamma=0.01, kernel=rbf; total time=   5.5s
[CV] END ...................C=1e-08, gamma=0.01, kernel=rbf; total time=   5.4s
[CV] END ...................C=1e-08, gamma=0.01, kernel=rbf; total time=   5.6s
```

Figure 6.10: Optimisation of the SVM model through grid search

```
lr_parameters = {
    'penalty':['elasticnet'],
    'C':[0.000001, 0.0001, 0.01, 1, 100],
    'solver':['saga'],
    'l1_ratio':[0.1, 0.2, 0.3, 0.4, 0.5 , 0.6, 0.7, 0.8],
    'max_iter':[10000],
    'multi_class':['auto']
}

folds = StratifiedKFold(n_splits=5, shuffle=True, random_state=5)

lr = LogisticRegression()
gridsearch_lr_elasticnet = GridSearchCV(lr, lr_parameters, cv=folds, verbose=2, scoring='f1_macro').fit(features, labels)
```

```
Fitting 5 folds for each of 40 candidates, totalling 200 fits
[CV] END C=1e-06, l1_ratio=0.1, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.1, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.1, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.1, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.1, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.2, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.2, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.2, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.2, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.2, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.3, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.3, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.3, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.3, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.3, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.4, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.4, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
[CV] END C=1e-06, l1_ratio=0.4, max_iter=10000, multi_class=auto, penalty=elasticnet, solver=saga; total time=   0.2s
```

Figure 6.11: Optimisation of the LR model using Elastic-Net through grid search

```
lr_parameters = {
    'solver':['liblinear'],
    'penalty':['l1','l2'],
    'C':[0.00000001, 0.000001, 0.0001, 0.01, 1, 100, 10000, 1000000, 100000000]
}

folds = StratifiedKFold(n_splits=5, shuffle=True, random_state=5)

lr = LogisticRegression()
gridsearch_lr = GridSearchCV(lr, lr_parameters, cv=folds, verbose=2, scoring='f1_macro').fit(features, labels)
```

```
Fitting 5 folds for each of 18 candidates, totalling 90 fits
[CV] END ..............C=1e-08, penalty=l1, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-08, penalty=l1, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-08, penalty=l1, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-08, penalty=l1, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-08, penalty=l1, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-08, penalty=l2, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-08, penalty=l2, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-08, penalty=l2, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-08, penalty=l2, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-08, penalty=l2, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-06, penalty=l1, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-06, penalty=l1, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-06, penalty=l1, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-06, penalty=l1, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-06, penalty=l1, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-06, penalty=l2, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-06, penalty=l2, solver=liblinear; total time=   0.0s
[CV] END ..............C=1e-06, penalty=l2, solver=liblinear; total time=   0.0s
```

Figure 6.12: Optimisation of the LR model using Lasso (L1) and Ridge (L2) through grid search

```
rf_parameters = {
    'max_depth': [10, 20, 30, 40, 50],
    'max_features': ['auto', 'sqrt'],
    'min_samples_leaf': [1, 4],
    'min_samples_split': [2, 5, 10],
    'n_estimators': [200, 400, 600, 800, 1000]
}

folds = StratifiedKFold(n_splits=5, shuffle=True, random_state=5)

rf = RandomForestClassifier()
gridsearch_rf = GridSearchCV(rf, rf_parameters, cv=folds, verbose=2, scoring='f1_macro').fit(features, labels)
```

```
Fitting 5 folds for each of 300 candidates, totalling 1500 fits
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=200; total time=   1.0s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=200; total time=   1.1s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=200; total time=   1.1s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=200; total time=   1.1s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=200; total time=   1.1s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=400; total time=   2.1s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=400; total time=   2.1s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=400; total time=   2.2s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=400; total time=   2.2s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=400; total time=   2.2s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=600; total time=   3.3s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=600; total time=   3.2s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=600; total time=   3.4s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=600; total time=   3.3s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=600; total time=   3.2s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=800; total time=   4.4s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=800; total time=   4.3s
[CV] END max_depth=10, max_features=auto, min_samples_leaf=1, min_samples_split=2, n_estimators=800; total time=   4.5s
```

Figure 6.13: Optimisation of the RF model through grid search

Upon the completion of the grid search optimisation, the set of parameters that yields the highest F-Measure score is returned. Table 6.1 shows the best parameters for each model and the corresponding F-measure before and after optimisation.

Table 6.1: Best parameters of each model

Models	Best parameters	F-measure after optimisation	F-measure before optimisation
SVM	• Kernel: ‘rbf’ • Gamma: 0.01 • C: 100000000	0.907	0.906
Elastic-Net LR	• Penalty: 'elasticnet' • C: 100 • Solver: ‘saga’ • L1-ratio: 0.6 • Max iteration: 10000	0.905	0.903
L1 and L2 LR	• Penalty: ‘l2’ • C: 1 • Solver: ‘liblinear’	0.900	0.903
RF	• Max depth: 10 • Max features: 'auto' • Minimum samples for a leaf: 1 • Minimum samples for a split: 10 • Number of estimators: 200	0.905	0.891

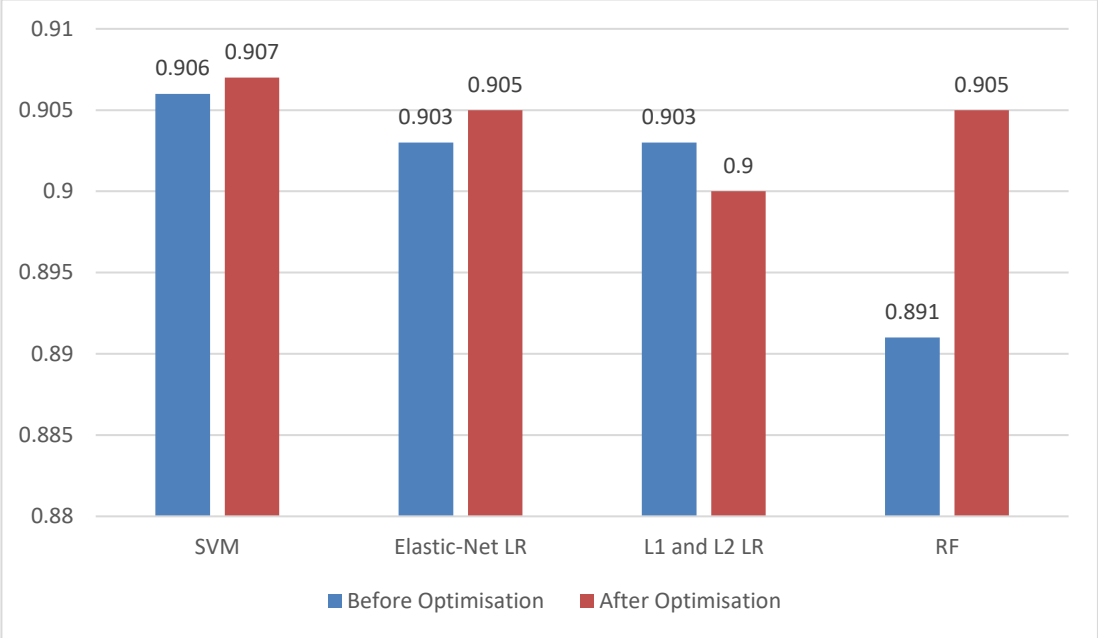


Figure 6.14: Comparison of F-measure before and after grid search optimisation

Overall, the improve in the F-measure is minimal all across the models with the most improved model being RF (1.4%), followed by Elastic-Net LR (0.2%), SVM (0.1%) and lastly, Lasso and Ridge LR (-0.3%) which showed degradation in F-measure. The changes in F-measures after performing grid search optimisation seem insignificant, which could be due to random variations.

However, the SVM model managed to maintain as the best performing data mining algorithm among the top three. Therefore, the SVM model instance with its set of best parameters was chosen to be deployed into the web-based system for classifying the conference papers. This was done through the Pickle library, which was used to serialise the TF-IDF vectoriser and SVM model instances into a file named 'models.p' which is then exported into the web application's directory.

```
# save vectoriser and trained SVM model
import pickle

svm_classifier = gridsearch_svm.best_estimator_

pickle = {
    'tf_idf_vectoriser': tfidf,
    'svm_classifier': svm_classifier
}
pickle.dump( pickles, open( 'models' + ".p", "wb" ) )
```

Figure 6.15: Serialising the TF-IDF vectoriser and model instances into a file

The file was then loaded into the web application through a configuration file in the web application as shown in Figure 6.15. The TF-IDF vectoriser and SVM model instances can now be readily used as methods by calling them through the ConferenceConfig class.

```
from django.apps import AppConfig
from django.conf import settings
import os
import pickle


class ConferenceConfig(AppConfig):
    name = 'conference'

    # create path to models
    path = os.path.join(settings.MODELS, 'models.p')

    # load models into separate variables
    # these will be accessible via this class
    with open(path, 'rb') as pickled:
        data = pickle.load(pickled)
    td_idf_vectoriser = data['tf_idf_vectoriser']
    svm_classifier = data['svm_classifier']
```

Figure 6.16: De-serialising the TF-IDF vectoriser and model instances

6.2 Web Application Implementation with Django

As mentioned earlier in Chapter 5, the project's web application is implemented with Django, which is a Python-based web framework based on a Model-View-Template (MVT) design. Therefore, in this section, the mappings of models, views and templates created in this project, are listed in this section.

6.2.1 Model

The models are defined classes in the model.py file. Each model has its fields and the corresponding database table that is mapped to the model.

Table 6.2: Models description

Model	Fields	Database table	Description
Chairman	- user	conference_chairman	The model that represents a chairman

			user.
Reviewer	- user - is_invited - domain	conference_reviewer	The model that represents a reviewer user.
Author	- user	conference_author	The model that represents an author user.
Conference	- title - description - created_by - start_date - end_date - submission_deadline - review_deadline - publish_review	conference_conference	The model that represents a conference. Holds foreign key to the chairman who created the conference.
Paper	- title - abstract - submit_date - conference - author - reviews - assigned_reviewers - decision - pdf - predicted_topic - corrected_topic - camera_ready	conference_paper	The model that represents a paper. Holds foreign key to the author who submitted the paper.
Review	- paper - reviewer - acceptance - comment	conference_review	The model that represents a review. Holds the foreign key to the paper that contains this review and the reviewer who gave this review.

6.2.2 View

The views are defined as classes in the view.py file. Each view may correspond to one or more models and the actions that are performed based on the application logic.

Table 6.3: Views description

View	Associated model	URL route	Description
Common views			
ConfLogin	Conference	accounts/login/	View that contains the authentication logic and query of a list of conferences.

ConferenceList	Conference	accounts/login/view_conferences/	View that contains the query to list all conferences.
PaperList	Paper	accounts/login/view_papers /<int:conference_id>	View that queries all the papers under a conference. For an author, only his own papers are queried.

PaperDetail	Paper, Review	accounts/login/view_paper_detail /<int:pk>	View that queries the details of a paper and its reviews. POST requests are also handled according to permissions. For a chairman, the actions of correcting topic, making chair decision and assigning reviewers are allowed. For a reviewer, giving a review is allowed. For an author, uploading the camera-ready paper is allowed once the paper is accepted. The reviews are also hidden from the author if the chairman has not published the reviews.
Chairman's views			
ConferenceForm	Conference	accounts/login/create_conference/	View that handles a POST request to create a conference
ReviewerList	Reviewer	accounts/login/view_reviewers/	View that queries all the reviewers. The chairman may also update whether a reviewer is invited or not.

RecorrectTopic	Paper	accounts/login/recorrect_topic /<int:pk>	View that handles a POST request to recorrect the predicted topic.
AssignPaper	Paper, Reviewer	accounts/login/assign_reviewers /<int:pk>	View that handles a POST request to assign a reviewer to a paper
Author's views			
PaperForm	Paper	accounts/login/create_paper/	View that handles a POST request to submit a paper. The topic prediction is also performed in this view upon the POST request.
SubmissionList	Paper	accounts/login/view_submissions/	View that queries all the submissions of an author.
Reviewer's views			
AssignedList	Paper	accounts/login/assigned_papers/	View that queries all the assigned papers of a reviewer.
ReviewerUpdate	Reviewer	accounts/login/update_domain/	View that handles a POST request for a reviewer to update his domain expertise.

6.2.3 Template

The templates are HTML files containing the UI designs under the template directory. They are also embedded with Python code in order to display data that is received from the views and perform minor logics. Each template is associated to at least one view that calls the template to be rendered to the client.

Table 6.4: Templates description

Template	Associated view
base.html	Non. It contains the base design that is inherited by all other templates.
login.html	ConfLogin
create_conference.html	ConferenceForm
conference_list.html	ConferenceList
create_paper.html	PaperForm
submission_list.html	SubmissionList
assigned_list.html	AssignedList
paper_list.html	PaperList
update_domain.html	ReviewerUpdate
reviewer_list.html	ReviewerList
paper_detail.html	PaperDetail
assign_reviewers.html	AssignPaper
recorrect_topic.html	RecorrectTopic

CHAPTER 7
TESTING

7.1 Unit Testing

Unit tests were performed for every unit of feature that is completed during the development phase. Performing unit tests ensure that every unit of feature or module behaves as intended according to the requirements.

7.1.1 Admin Login Module

Table 7.1: Unit test for Admin Login module

Test Case	Execution Steps	Expected Result	Pass/Fail
Login with empty fields.	1. Login without any field entries.	Prompted to enter empty fields.	Pass
Login with wrong credentials.	1. Login with credentials that does not belong to an admin's account.	Login failed with incorrect username and password.	Pass
Login with correct credentials	1. Login with an admin's credentials.	Login to admin page successfully.	Pass

7.1.2 Account Creation by Admin Module

Table 7.2: Unit test for Account Creation by Admin module

Test Case	Execution Steps	Expected Result	Pass/Fail
Add an author user	1. Click add a user. 2. Enter a new username and password. 3. Confirm the password. 4. Enter the user's first name and last name. 5. Select the author user group. 6. Create an author.	User with an author group and class is added in the user and author pages.	Pass

	7. Assign the user as an author.		
Add a chairman user	1. Click add a user. 2. Enter a new username and password. 3. Confirm the password. 4. Enter the user's first name and last name. 5. Select the chairman user group. 6. Create a chairman. 7. Assign the user as a chairman.	User with a chairman group and class is added in the user and chairman pages.	Pass
Add a reviewer user	1. Click add a user. 2. Enter a new username and password. 3. Confirm the password. 4. Enter the user's first name and last name. 5. Select the reviewer user group. 6. Create a reviewer. 7. Assign the user as a reviewer.	User with a reviewer group and class is added in the user and reviewer pages.	Pass

7.1.3 User Login Module

Table 7.3: Unit test for User Login module

Test Case	Execution Steps	Expected Result	Pass/Fail
Login with empty fields.	1. Login without any field entries.	Prompted to enter empty fields.	Pass
Login with wrong credentials.	1. Login with credentials that does not belong to a chairman, reviewer or author.	Login failed with incorrect username and password.	Pass
Login with correct credentials	1. Login with a chairman, reviewer or author credentials.	Login to conferences page successfully.	Pass

7.1.4 Create Conference Module

Table 7.4: Unit test for Create Conference module

Test Case	Execution Steps	Expected Result	Pass/Fail
Create a conference with empty fields.	1. Create a conference without any entries.	Prompted to enter empty fields.	Pass
Create a conference with invalid dates.	1. Fill in the title and description. 2. Select the end date / submission deadline / review deadline to be earlier than the present date.	Invalid date message prompted to user.	Pass
Create a conference with valid dates	1. Fill in the title and description. 2. Select the end date to be after the present date.	Conference successfully created and listed in the conferences page.	Pass

	3. Select the review deadline to be between the present date and end date. 4. Select the submission deadline to be between the present date and the review deadline.		

7.1.5 Invite Reviewers Module

Table 7.5: Unit test for Invite Reviewers module

Test Case	Execution Steps	Expected Result	Pass/Fail
Invite a reviewer	1. Click on the 'invite reviewer' button for a reviewer that is not yet invited.	'Invited for review' status of the reviewer changes from No to Yes.	Pass
Disallow a reviewer	1. Click on the 'disallow reviewer' button for a reviewer that already invited.	'Invited for review' status of the reviewer changes from Yes to No.	Pass

7.1.6 Publish Reviews Module

Table 7.6: Unit test for Publish Reviews module

Test Case	Execution Steps	Expected Result	Pass/Fail
Allow reviews to be seen by authors.	1. Toggle the Yes/No button to Yes.	Toggled to Yes.	Pass
Do not allow reviews to be seen by authors.	1. Toggle the Yes/No button to No.	Toggled to No.	Pass

7.1.7 Assign Reviewers Module

Table 7.7: Unit test for Publish Reviews module

Test Case	Execution Steps	Expected Result	Pass/Fail
Assign a reviewer	1. Click on the 'Assign Reviewers' button. 2. Select any of the reviewers listed.	Reviewer is listed in the reviewer assignment table.	Pass

7.1.8 Download PDF Module

Table 7.8: Unit test for Download PDF module

Test Case	Execution Steps	Expected Result	Pass/Fail
Download the PDF file of a paper	1. Click on the 'Download' button.	The PDF file is downloaded.	Pass

7.1.9 Chair Decision Module

Table 7.9: Unit test for Chair Decision module

Test Case	Execution Steps	Expected Result	Pass/Fail
Accepting a paper	1. Select 'Accept' in the dropdown list. 2. Click 'Submit Decision' button.	The 'Chair Decision' changes from None to Accept.	Pass
Rejecting a paper	1. Select 'Reject' in the dropdown list. 2. Click 'Submit Decision' button.	The 'Chair Decision' changes from None to Reject.	Pass

7.1.10 Paper Submission Module

Table 7.10: Unit test for Paper Submission module

Test Case	Execution Steps	Expected Result	Pass/Fail
Invalid submission	1. Leave all or any of the fields empty. 2. Click the 'Submit' button.	Invalid fields message prompted to user.	Pass
Valid submission	1. Fill in the title. 2. Fill in the abstract. 3. Select the conference. 4. Upload the PDF file. 5. Click the 'Submit' button.	The submission is successful and recorded in the author's submitted papers and the predicted topic is set in the paper details.	Pass

7.1.11 Camera-ready Submission Module

Table 7.11: Unit test for Camera-ready Submission module

Test Case	Execution Steps	Expected Result	Pass/Fail
Invalid submission	1. Click on the 'Submit Camera Ready' button without any uploaded files	Invalid fields message prompted to user.	Pass
Valid submission	1. Upload the camera-ready PDF file. 2. Click the 'Submit Camera Ready' button.	The submission is successful, and a 'camera-ready' tag is shown on the file name.	Pass

7.1.12 Update Domain Expertise Module

Table 7.12: Unit test for Update Domain Expertise module

Test Case	Execution Steps	Expected Result	Pass/Fail
Select a domain expertise	1. Select current domain expertise to another domain expertise.	The domain expertise is updated.	Pass

7.1.13 Give Review Module

Table 7.13: Unit test for Give Review module

Test Case	Execution Steps	Expected Result	Pass/Fail
Give an invalid review	1. Leave one of the entries empty. 2. Click the 'Add Review' button.	Invalid fields message prompted to user.	Pass
Give a valid review	1. Fill in the comment. 2. Select an 'Acceptance' option.	Review is added into the reviews section.	Pass

7.2 Integration Testing

After unit testing, integration testing was performed on modules that are preceding each other. Integration testing is to detect for any errors when all the modules tested in the unit testing were integrated when actions flow from one module to another.

7.2.1 Account Creation by Admin and User Login

Table 7.14: Integration test for Account Creation by Admin and User Login modules

Test Case	Create accounts and login by user successfully.
Test Procedure	1. Login to admin. 2. Create chairman, reviewer and author accounts as an admin. 3. Login as a chairman, reviewer and author using the credentials created by the admin.
Expected Results	Accounts are created successfully, and chairman, reviewer and author users were able to login.
Pass/Fail	Pass

7.2.2 Create Conference and Paper Submission

Table 7.15: Integration test for Create Conference and Paper Submission modules

Test Case	Create a conference and submit a paper to the conference successfully.
Test Procedure	1. Login as a chairman 2. Fill in the conference creation fields and create a conference. 3. Login as an author. 4. Fill in the paper submission fields and submit a paper to the created conference.
Expected Results	The conference is created successfully, and the paper could be submitted under the conference.
Pass/Fail	Pass

7.2.3 Invite Reviewer and Assign Reviewer

Table 7.16: Integration test for Invite Reviewer and Assign Reviewer modules

Test Case	Invite a reviewer and assign the reviewer to a paper.
Test Procedure	1. Login as a chairman 2. Click the 'Invite Reviewer' button to invite a reviewer. 3. Select a paper and click the 'Assign Reviewers' button. 4. Select the reviewer that was invited. 5. Click the 'Assign Reviewers' button.
Expected Results	As a chairman, the reviewer could be invited and assigned to the paper successfully.
Pass/Fail	Pass

7.2.4 Give Review and Submit Chair Decision

Table 7.17: Integration test for Give and Submit Chair Decision modules

Test Case	Give a review to a paper and submit a chair decision for the paper.
Test Procedure	1. Login as a reviewer. 2. Select a paper and fill in the review fields. 3. Click the 'Add Review' button. 4. Login as a chairman. 5. Select the similar paper that was reviewed. 6. Select a decision for the chair's decision field. 7. Click the 'Submit Chair Decision' button.
Expected Results	As a reviewer, the review could be successfully added into the reviews section. As a chairman the chair decision is shown on the paper details after submitting the decision.
Pass/Fail	Pass

7.2.5 Camera-ready Submission and Download PDF

Table 7.18: Integration test for Camera-ready Submission and Download PDF modules

Test Case	Perform submission of the camera-ready paper as an author for an accepted paper and download the PDF file of the paper.
Test Procedure	1. Login as a reviewer. 2. Select a paper that was accepted by the chair. 3. Click the ‘Choose File’ button. 4. Select a PDF file to be uploaded from the local system. 5. Click the ‘Submit Camera Ready’ button. 6. Click the ‘Download’ button for the submitted paper.
Expected Results	As an author, a camera-ready PDF file could be uploaded and submitted. Then, the similar PDF file could be downloaded as well.
Pass/Fail	Pass

7.3 System Testing

After completing integration testing, it is then followed by system testing to ensure that all the modules as a whole are well integrated and all processes can be performed in the system.

7.3.1 Paper Assignment Process

Table 7.19: System test for paper assignment process.

Test Case	Full process of paper assignment.
Test Procedure	1. Login as a chairman. 2. Select ‘Create Conference’ and fill in the fields to create a conference. 3. Login as an author. 4. Select ‘Submit Paper’ and fill in the fields to submit a paper under the conference created by the chairman. 5. Login as a chairman. 6. Select ‘Invite Reviewers’ and click the ‘Invite Reviewer’ button for a reviewer. 7. Login as the invited reviewer. 8. Click ‘Update Domain Expertise’ and intentionally select a domain expertise that matches with the submitted paper’s predicted topic. 9. Login as the chairman. 10. Click into the submitted paper and click the ‘Assign Reviewers’ button. 11. Check that the invited reviewer is listed as a recommended reviewer with matching domain expertise with the paper’s predicted topic. 12. Select the reviewer and click the ‘Assign Reviewers’ button.
Expected Results	1. Logged in successfully as a chairman, reviewer and author. 2. Created the conference successfully as a chairman. 3. Submitted the paper successfully as an author and predicted topic was generated.

	4. Invited a reviewer successfully as a chairman. 5. Updated domain expertise successfully as a reviewer. 6. Assigned the reviewer successfully as an author and the reviewer was in the recommended list.
Pass/Fail	Pass

7.3.2 Paper Review Process

Table 7.20: System test for paper review process.

Test Case	Full process of paper review.
Test Procedure	1. Login as a reviewer. 2. Select ‘Assigned Papers’ and click into one of the assigned papers. 3. Click ‘Download’ button to download the PDF file. 4. Fill in the comment field and select an acceptance option. 5. Click ‘Add Review’ button to submit the review. 6. Login as a chairman 7. Select the paper that was reviewed by the reviewer. 8. Check that the review given by the reviewer can be seen. 9. Accept the paper by selecting ‘Accept’ decision and click the ‘Submit Decision’ button. 10. Login as the author who submitted the paper. 11. Check that the decision of the paper is now ‘Accept’. 12. Select a PDF file to be uploaded as the camera-ready paper. 13. Click the ‘Submit Camera Ready’ button and check that the PDF file name is tagged with ‘camera-ready version’.
Expected Results	1. Login successfully as chairman, reviewer and author. 2. As a reviewer, the PDF file of the paper could be downloaded. 3. As a reviewer, a review could be given to the reviews section of the paper. 4. As a chairman, the review given by the reviewer can be seen

	for the paper. 5. As a chairman, the decision for the paper could be made. 6. As an author, the camera-ready PDF file can be uploaded and submitted when the paper is accepted by the chairman.
Pass/Fail	Pass

CHAPTER 8
CONCLUSION AND RECOMMENDATION

8.1 Introduction

In this chapter, the benefits, limitations, future improvements, and recommendation of the project are discussed, followed by the conclusion.

8.2 Benefits of the Project

In much hopefulness, the implementation of this project will be able to contribute to the research domain, especially in the area of conference proceedings. The automated classification system in this project is able to perform topic predictions for AI related conference papers, at 0.907 F-measure which is equivalent to 90.7% accurateness.

Undoubtedly, it will be able to help the chairmen in their paper assigning process by reducing the time taken to assign papers to reviewers. Besides that, the web application also contains a suite of features for managing a research conference which digitalises research conference processes such as conference creation, paper submission, paper assignment, paper review and paper acceptance processes.

8.3 Limitations of the Project

Every project is bound to have its limitation. In this project, the limitations are identified as follows:

- Classification of research papers is limited to AI topics only.
- Optimisation of the data mining algorithms were not able to yield significant improvements in F-measure evaluation.

8.4 Recommendations for Future Improvements

For future works, the following are some recommendations that can be considered to improve the project:

- Semi-supervised approach to generate research topics automatically

  The use of unsupervised and supervised learning approaches can be combined as a semi-supervised approach to generate research topics automatically. Clustering, which is an unsupervised learning technique, can be used to find common keywords in collections of papers which are identified as clusters. Each cluster of papers can then be labelled by the common topic, which can then be used as a data set to train a supervised learning algorithm to predict new papers according to the topics found by the clustering approach.

- Perform grid search optimisation on different combinations of pre-processing and transformation techniques

  The reason grid search optimisation of the data mining algorithms yielded insignificant improvement could be due to the complexity of the data set since AI topics may share many keywords which are harder to classify. Performing grid search optimisation on different combinations of pre-processing and transformation techniques could discover the best set of techniques to better represent the words as numerical features.

8.5 Conclusion

In conclusion, the project objectives were met. An automated web-based conference paper system for researchers was developed to facilitate their research conference management activities. The manual process of assigning papers to reviewers by using classification models was also achieved by deploying the model into the web-based conference paper system to be used by chairmen. Following the KDD process as an empirical approach, the SVM model was able to be to be selected as the best classification model (0.907 F-measure) through evaluation of results for the conference paper system.

REFERENCES

Abd, T., Abd, M., & Kholief, S. (2016). Identify and Classify Critical Success Factor of Agile Software Development Methodology Using Mind Map. *International Journal of Advanced Computer Science and Applications*, *7*(5), 83–92. https://doi.org/10.14569/ijacsa.2016.070513

Aggarwal, C. C., & Zhai, C. X. (2012). A survey of text classification algorithms. In *Mining Text Data*. https://doi.org/10.1007/978-1-4614-3223-4_6

Arusada, M. D. N., Putri, N. A. S., & Alamsyah, A. (2017). Training data optimisation strategy for multiclass text classification. *2017 5th International Conference on Information and Communication Technology, ICoIC7 2017*, *0*(c). https://doi.org/10.1109/ICoICT.2017.8074652

Banga, R., Bhardwaj, A., Peng, S.-L., & Shrivastava, G. (2018). *Authorship Attribution for Online Social Media* (pp. 141–165). https://doi.org/10.4018/978-1-5225-5097-6.ch008

Chandra Mohan, B., & Baskaran, R. (2012). A survey: Ant Colony Optimization based recent research and implementation on several engineering domain. In *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2011.09.076

Charlin, L., & Zemel, R. S. (2013). The Toronto Paper Matching System: An automated paper-reviewer assignment system. *Icml*, *28*. http://cmt.research.microsoft.com/cmt/

De Mántaras, R. L. (1991). A Distance-Based Attribute Selection Measure for Decision Tree Induction. *Machine Learning*. https://doi.org/10.1023/A:1022694001379

Dehnad, K. (1989). A simple way of improving the login security. *Computers and Security*, *8*(7), 607–611. https://doi.org/10.1016/0167-4048(89)90053-9

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(9–10), 833–859. https://doi.org/10.1016/j.infsof.2008.01.006

Eberlein, A. (2003). Requirements Engineering and Agile Software Development, University of Calgary. *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.*, 1–6.

Fan, R., Wang, X., Lin, C., Chang, K.-W., Hsieh, C.-J., Wang, X., & Lin, C. (2014).

LIBLINEAR : A Library for Large Linear Classification. *Journal of Machine Learning Research*.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Communications of the ACM*, *39*(11), 27–34. https://doi.org/10.1145/240455.240464

Giovanelli, C., Liu, X., Sierla, S., Vyatkin, V., & Ichise, R. (2017). Towards an aggregator that exploits big data to bid on frequency containment reserve market. *Proceedings IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*. https://doi.org/10.1109/IECON.2017.8217316

Gupta, B., Gupta, O., & Sawhney, B. K. (2013). A Framework for Conference Management System. *International Journal of Management & Information Technology*, *5*(2), 483–487. https://doi.org/10.24297/ijmit.v5i2.764

Huang, K. (2015). Unconstrained smartphone sensing and empirical study for sleep monitoring and self-management. In *ProQuest Dissertations and Theses*.

Ikonomakis, M., Kotsiantis, S., & Tampakas, V. (2005). Text classification using machine learning techniques. *WSEAS Transactions on Computers*, *4*(8), 966–974. https://doi.org/10.11499/sicejl1962.38.456

Instructions for the Preparation of a Camera-Ready Manuscript. (n.d.). *Nomen NESCIO*, *244236*, 244236.

Isa, D., Lee, L. H., Kallimani, V. P., & Rajkumar, R. (2008). Text document preprocessing with the bayes formula for classification using the support vector machine. *IEEE Transactions on Knowledge and Data Engineering*, *20*(9), 1264–1272. https://doi.org/10.1109/TKDE.2008.76

James, S. (2007). *The Art of Agile Development: Pragmatic Guide to Agile Software Development*. O'Reilly Media, Inc.

Juan, A., & Vidal, E. (2002). On the use of Bernoulli mixture models for text classification. *Pattern Recognition*. https://doi.org/10.1016/S0031-3203(01)00242-4

Kowsari, K., Brown, D. E., Heidarysafa, M., Jafari Meimandi, K., Gerber, M. S., & Barnes, L. E. (2017). HDLTex: Hierarchical Deep Learning for Text Classification. *Proceedings - 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017*. https://doi.org/10.1109/ICMLA.2017.0-134

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information (Switzerland)*, *10*(4), 1–68. https://doi.org/10.3390/info10040150

Liu, Y., Bi, J. W., & Fan, Z. P. (2017). Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms. *Expert Systems with Applications*, *80*, 323–339. https://doi.org/10.1016/j.eswa.2017.03.042

M, H., & M.N, S. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), 01–11. https://doi.org/10.5121/ijdkp.2015.5201

Mustafa, G. (2020). *Multi-label Classification of Research Articles using Word2Vec and Identification of Similarity Threshold*.

Nguyen, J., Sanchez-Hernandez, G., Agell, N., Rovira, X., & Angulo, C. (2016). An OWA ased multi-criteria system for assigning papers to reviewers. *Frontiers in Artificial Intelligence and Applications*. https://doi.org/10.3233/978-1-61499-696-5-253

OpenConf. (2020). Zakon Group LLC. https://www.openconf.com/

Quinlan, J. R. (1987). Simplifying decision trees. *International Journal of Man-Machine Studies*. https://doi.org/10.1016/S0020-7373(87)80053-6

Sergienko, R., Shan, M., & Minker, W. (2016). A comparative study of text preprocessing approaches for topic detection of user utterances. *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, 1826–1831.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Sun, Z., Bebis, G., & Miller, R. (2002). Improving the performance of on-road vehicle detection by combining Gabor and wavelet features. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*. https://doi.org/10.1109/ITSC.2002.1041201

Uysal, A. K., & Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing and Management*. https://doi.org/10.1016/j.ipm.2013.08.006

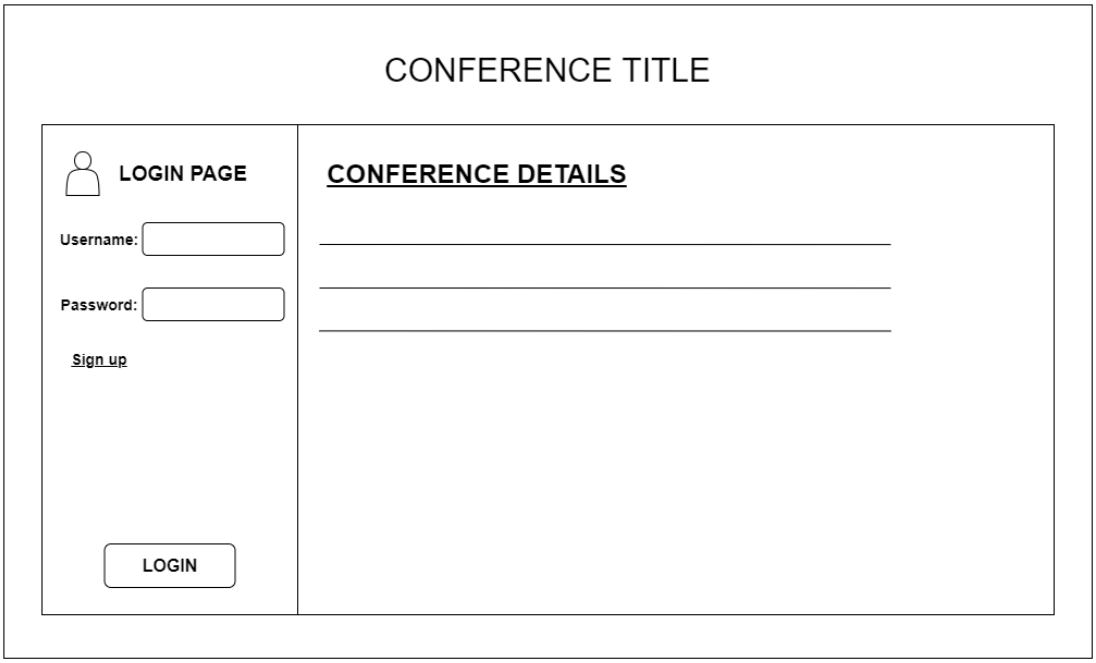
Vijayan, V. K., Bindu, K. R., & Parameswaran, L. (2017). A comprehensive study of text classification algorithms. *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017, 2017-Janua*, 1109–1113. https://doi.org/10.1109/ICACCI.2017.8125990

Vo, D. T., & Ock, C. Y. (2015). Learning to classify short text from scientific documents using topic models with various types of knowledge. *Expert Systems with Applications*, *42*(3), 1684–1698. https://doi.org/10.1016/j.eswa.2014.09.031

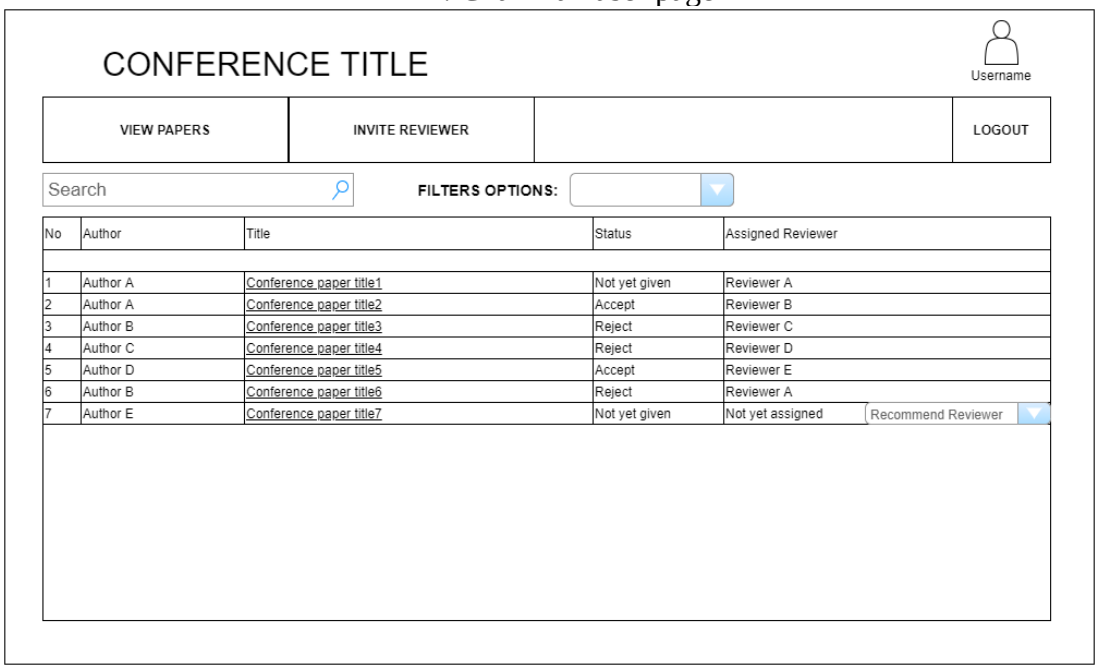
Wu, T. F., Lin, C. J., & Weng, R. C. (2004). Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*.

APPENDICES

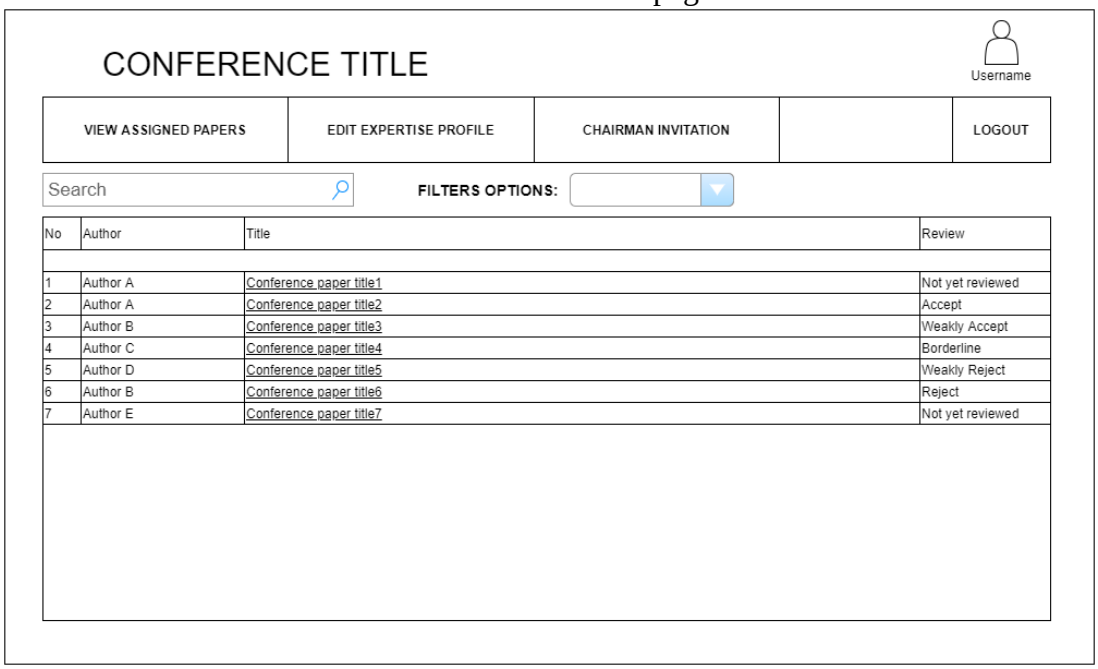
APPENDIX A: Login page



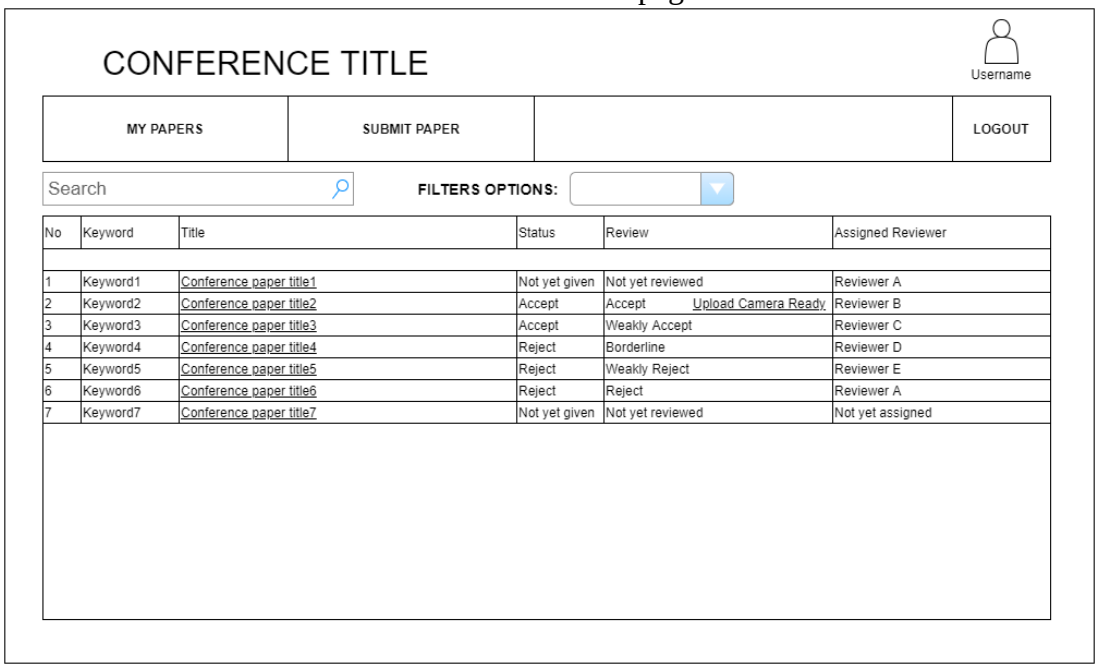
APPENDIX B: Chairman user page



APPENDIX C: Reviewer user page



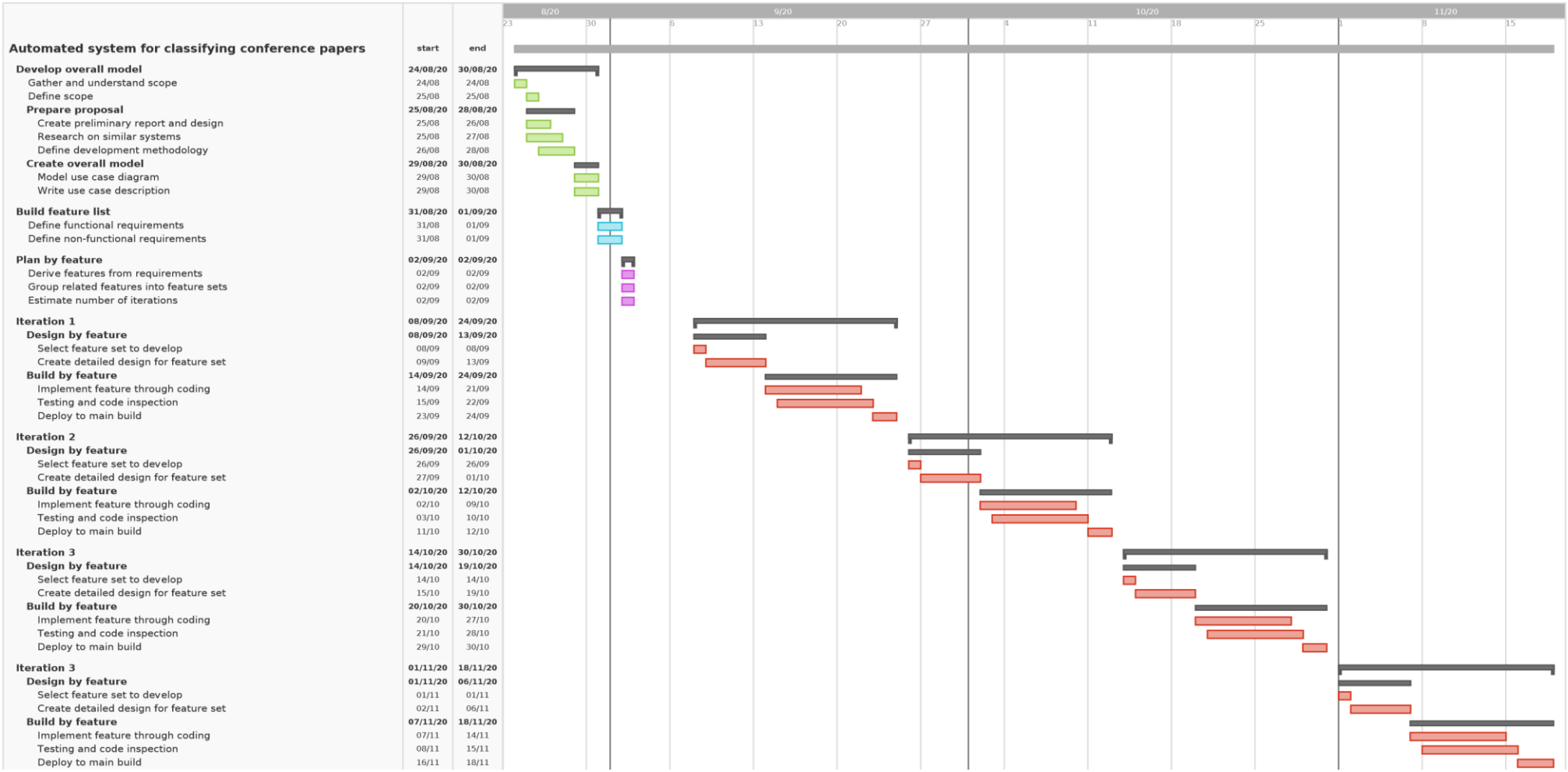
APPENDIX D: Author user page



APPENDIX E: Work Breakdown Structure

Automated system for classifying conference papers	start	end
Develop overall model	**24/08/20**	**30/08/20**
Gather and understand scope	24/08	24/08
Define scope	25/08	25/08
Prepare proposal	**25/08/20**	**28/08/20**
Create preliminary report and design	25/08	26/08
Research on similar systems	25/08	27/08
Define development methodology	26/08	28/08
Create overall model	**29/08/20**	**30/08/20**
Model use case diagram	29/08	30/08
Write use case description	29/08	30/08
Build feature list	**31/08/20**	**01/09/20**
Define functional requirements	31/08	01/09
Define non-functional requirements	31/08	01/09
Plan by feature	**02/09/20**	**02/09/20**
Derive features from requirements	02/09	02/09
Group related features into feature sets	02/09	02/09
Estimate number of iterations	02/09	02/09
Iteration 1	**08/09/20**	**24/09/20**
Design by feature	**08/09/20**	**13/09/20**
Select feature set to develop	08/09	08/09
Create detailed design for feature set	09/09	13/09
Build by feature	**14/09/20**	**24/09/20**
Implement feature through coding	14/09	21/09
Testing and code inspection	15/09	22/09
Deploy to main build	23/09	24/09
Iteration 2	**26/09/20**	**12/10/20**
Design by feature	**26/09/20**	**01/10/20**
Select feature set to develop	26/09	26/09
Create detailed design for feature set	27/09	01/10
Build by feature	**02/10/20**	**12/10/20**
Implement feature through coding	02/10	09/10
Testing and code inspection	03/10	10/10
Deploy to main build	11/10	12/10
Iteration 3	**14/10/20**	**30/10/20**
Design by feature	**14/10/20**	**19/10/20**
Select feature set to develop	14/10	14/10
Create detailed design for feature set	15/10	19/10
Build by feature	**20/10/20**	**30/10/20**
Implement feature through coding	20/10	27/10
Testing and code inspection	21/10	28/10
Deploy to main build	29/10	30/10
Iteration 3	**01/11/20**	**18/11/20**
Design by feature	**01/11/20**	**06/11/20**
Select feature set to develop	01/11	01/11
Create detailed design for feature set	02/11	06/11
Build by feature	**07/11/20**	**18/11/20**
Implement feature through coding	07/11	14/11
Testing and code inspection	08/11	15/11
Deploy to main build	16/11	18/11

APPENDIX F: Gantt Chart



Automated system for classifying conference papers	start	end
Develop overall model	**24/08/20**	**30/08/20**
Gather and understand scope	24/08	24/08
Define scope	25/08	25/08
Prepare proposal	**25/08/20**	**28/08/20**
Create preliminary report and design	25/08	26/08
Research on similar systems	25/08	27/08
Define development methodology	26/08	28/08
Create overall model	**29/08/20**	**30/08/20**
Model use case diagram	29/08	30/08
Write use case description	29/08	30/08
Build feature list	**31/08/20**	**01/09/20**
Define functional requirements	31/08	01/09
Define non-functional requirements	31/08	01/09
Plan by feature	**02/09/20**	**02/09/20**
Derive features from requirements	02/09	02/09
Group related features into feature sets	02/09	02/09
Estimate number of iterations	02/09	02/09
Iteration 1	**08/09/20**	**24/09/20**
Design by feature	**08/09/20**	**13/09/20**
Select feature set to develop	08/09	08/09
Create detailed design for feature set	09/09	13/09
Build by feature	**14/09/20**	**24/09/20**
Implement feature through coding	14/09	21/09
Testing and code inspection	15/09	22/09
Deploy to main build	23/09	24/09
Iteration 2	**26/09/20**	**12/10/20**
Design by feature	**26/09/20**	**01/10/20**
Select feature set to develop	26/09	26/09
Create detailed design for feature set	27/09	01/10
Build by feature	**02/10/20**	**12/10/20**
Implement feature through coding	02/10	09/10
Testing and code inspection	03/10	10/10
Deploy to main build	11/10	12/10
Iteration 3	**14/10/20**	**30/10/20**
Design by feature	**14/10/20**	**19/10/20**
Select feature set to develop	14/10	14/10
Create detailed design for feature set	15/10	19/10
Build by feature	**20/10/20**	**30/10/20**
Implement feature through coding	20/10	27/10
Testing and code inspection	21/10	28/10
Deploy to main build	29/10	30/10
Iteration 3	**01/11/20**	**18/11/20**
Design by feature	**01/11/20**	**06/11/20**
Select feature set to develop	01/11	01/11
Create detailed design for feature set	02/11	06/11
Build by feature	**07/11/20**	**18/11/20**
Implement feature through coding	07/11	14/11
Testing and code inspection	08/11	15/11
Deploy to main build	16/11	18/11