



Project Proposal

**An AI-Driven Framework for Gathering and Documenting Business
Requirements**

Prepared By

Name:Sahab AlHumaed

ID No.:251002651

**Submitted in fulfillment of the requirements for the degree Master
of Science in Artificial Intelligence**

At

Collage of Informatics

Midocean University UAE

Supervisor Name: Dr.Kahled Eskaf

**Collage of Informatics, Midocean University,
Comoros**

Email: 251002651@st.midocean.edu.km

1. Project Overview

The Business requirements document (BRD) is a major part of the software development life cycle (SDLC) process. Manual preparation of BRD is an extremely time-taking and effort-demanding process. An analyst has to collect those pieces of information embedded in meeting minutes, emails, rough notes, memos of multiple stakeholders of an organization to create a BRD. Manual process is error-prone and often results in loss of information/inconsistency information which may affect the quality of the product. This project aims to design a Smart BRD Generator that utilizes the power of Artificial Intelligence. This System will harness ML and NLP algorithms to automatically convert unstructured input to a structured Business Requirement Document.

The proposed solution will :

1. Identify and Extract Functional Requirements (FRs) and Non-Functional Requirements (NFRs).
2. Classify NFRs into several types like Security, Performance, Maintainability, Usability and Availability.
3. Mark Ambiguity, Raise questions for clarifications,
4. Automatically generate a professional BRD that follows industry standards.

The proposed framework builds on the latest research in this field. In recent years, several works have addressed the classification of requirements using machine learning and deep learning techniques. These studies have introduced advanced approaches such as hierarchical classification, zero-shot learning, and deep learning frameworks for the classification of non-functional requirements.

2. Project Goal

The main objectives of this project are to:

1. Develop an AI-driven tool to automate the capture, classification and documentation of business requirements.
2. Decrease the time and resources spent on BRD preparation.
3. Improve requirements documentation accuracy and consistency, avoiding project failure.

3. Approach

- **Data Collection:** The project will leverage publicly available datasets like PROMISE_exp, DOSSPRE, PURE, SecReq and custom datasets like meeting transcripts, emails, and project draft notes.
- **Data Processing:**
 - **Preprocessing:** tokenization, lemmatization, stopword removal, normalization, and punctuation filtering.
 - **Feature Representation:** using TF-IDF, Sentence-BERT embeddings, and domain-specific embeddings such as SObert.

- **Model Selection**
- **Machine Learning models:** Support Vector Machines (SVM), Logistic Regression, Naïve Bayes, and Random Forest.
- **Deep Learning models:** Artificial Neural Networks (ANN), BiLSTM, CNN, and ensemble classifiers.
- **Zero-Shot Learning:** using **Sentence-BERT** and **Bert4RE** to handle data-scarce situations.
- **Hybrid Approaches:** combining ML classifiers with **cosine similarity** for improved semantic understanding.

- **Testing and Evaluation**

- **Metrics:** Accuracy, Precision, Recall, and F1-score (both macro and micro averages).
- **Validation:** 10-fold cross-validation and project-based evaluation.
- **Benchmarking:** comparing results against existing methods reported in the literature.

4. Project Steps and Timeline

Phase	Task	Duration	Deliverable
Phase 1	Data Collection & Preparation	1 month	Cleaned, preprocessed datasets
Phase 2	Model Development & Training	2 months	Trained ML/DL models
Phase 3	Testing & Evaluation	1 month	Evaluation report
Phase 4	Prototype Development	2 months	Smart BRD Generator prototype
Phase 5	Final Adjustments & Documentation	1 month	Thesis + demonstration

5. Expected Results

- Enhanced precision in identifying and categorizing business requirements, including functional and non-functional, from unstructured data like emails, meeting minutes, and draft notes.
- Clearer insight into how to use AI and ML in RE, e.g. how to use deep learning and zero-shot learning for small / imbalanced datasets.
- A working AI that can automatically generate BRDs in a professional and standardized manner, with little human effort, to guide and help analysts make decisions.

6. Resources Needed

- **Software:** Python, TensorFlow or PyTorch, Scikit-learn, HuggingFace Transformers, Gradio (for deployment).
- **Datasets:** PROMISE_exp, DOSSPRE, PURE, SecReq, and other publicly available repositories.
- **Hardware:** GPU-enabled workstation (16–32 GB RAM, NVIDIA RTX 3060 or better).
- **Support:** Guidance from supervisor, access to computing labs, and input from subject-matter experts to validate generated BRDs.

7. References

- Bagies, T. (2024). Classifying software security requirements into confidentiality, integrity, and availability using machine learning approaches. *PeerJ Computer Science*, 10, e2554. <https://doi.org/10.7717/peerj-cs.2554>
- Sharma, N. (2020). *Requirements classification dataset* (V2). Mendeley Data. <https://data.mendeley.com/datasets/23xtbvk6yp/2>
- Guzmán, E., Azócar, D., & Li, Y. (2021). *PROMISE_exp: Dataset for requirements classification*. Zenodo. <https://doi.org/10.5281/zenodo.4530183>
- Mitrevski, A. (n.d.). *PROMISE_exp dataset* (.arff). GitHub. https://github.com/AleksandarMitrevski/se-requirements-classification/blob/master/0-datasets/PROMISE_exp/PROMISE_exp.arff
- Rahman, A., Nayem, A., & Siddik, S. (2023). Non-functional requirements classification using machine learning algorithms. *International Journal of Intelligent Systems and Applications*, 15(3), 56–69. <https://www.mecs-press.org/ijisa/ijisa-v15-n3/IJISA-V15-N3-5.pdf>
- Binkhonain, M., & Zhao, L. (2023). A machine learning approach for hierarchical classification of software requirements. *Machine Learning with Applications*, 12, 100457. <https://doi.org/10.1016/j.mlwa.2023.100457>
- Cleland-Huang, J., Settini, R., Zou, X., & Solc, P. (2007). Automated classification of non-functional requirements. *Requirements Engineering*, 12(2), 103–120. <https://doi.org/10.1007/s00766-007-0045-1>
- Houmb, S. H., Islam, S., Knauss, E., Jürjens, J., & Schneider, K. (2009). Eliciting security requirements and tracing them to design. *Requirements Engineering*, 15(1), 63–93. <https://doi.org/10.1007/s00766-009-0093-9>
- Dalpiaz, F., Dell’Anna, D., Aydemir, F. B., & Çevikol, S. (2019). Requirements classification with interpretable machine learning and dependency parsing. *2019 IEEE 27th International Requirements Engineering Conference (RE)*, 142–152. <https://doi.org/10.1109/RE.2019.00025>
- Ferrari, A., Spagnolo, G. O., & Gnesi, S. (2017). PURE: A dataset of public requirements documents. *2017 IEEE 25th International Requirements Engineering Conference (RE)*, 502–505. <https://zenodo.org/records/1414117>
- Firesmith, D. (2003). Engineering security requirements. *Journal of Object Technology*, 2(1), 53–68. <https://doi.org/10.5381/jot.2003.2.1.c6>
- Kadebu, P., Sikka, S., Tyagi, R. K., & Chiurunge, P. (2023). DOSSPRE – Dataset of Students’ Software Project Requirements. Mendeley Data. <https://data.mendeley.com/datasets/23xtbvk6yp/2>
- Martens, D., Méndez, D., Franch, X., & Vogelsang, A. (2023). Zero-Shot Learning for Requirements Classification: An Exploratory Study. *arXiv preprint*. <https://arxiv.org/abs/2302.04723>
- Mendes, E., & Mosley, N. (2006). *PROMISE NFR Dataset*. PROMISE Software Engineering Repository (mirror). <https://www.kaggle.com/datasets/iamsouvik/promise>
- Sharma, A., & Trivedi, R. (2018). *SecReq: A dataset for security requirements classification*. Zenodo. <https://zenodo.org/records/4530183>

- Dekhtyar, A., & Fong, V. (2017). *RE Data Challenge 2017 Dataset*. IEEE RE Conference. <https://doi.org/10.1109/RE.2017.26>
- Rahman, K., Ghani, A., Misra, S., & Rahman, A. U. (2024). A deep learning framework for non-functional requirement classification. *Scientific Reports*, 14, 3216. <https://doi.org/10.1038/s41598-024-52802-0>
- Khayashi, F., Jamasb, B., Akbari, R., & Shamsinejadbabaki, P. (2022). Deep Learning Methods for Software Requirement Classification: A Performance Study on the PURE dataset. *arXiv preprint*. <https://arxiv.org/abs/2211.05286>
- Hassan, S., Yasin, A., Li, Q., Aurangzeb, K., & Khan, J. A. (2024). A systematic mapping to investigate the application of machine learning techniques in requirement engineering activities. *CAAI Transactions on Intelligence Technology*, 9(6), 1412–1434. <https://ietresearch.onlinelibrary.wiley.com/doi/epdf/10.1049/cit2.12348>
- Rashme, T. Y. (2024). Mapping Software Requirements: An Overview of Classification Strategies. *International Journal of Applied Information Systems (IJ AIS)*, 12(45), 1–6. <https://www.ijais.org/archives/volume12/number45/rashme-2024-ijais-451976.pdf>
- Idate, S., Rao, T. S., Gayakwad, M., Paygude, P., Chavan, P., Pawar, R., & Kadam, K. (2024). Performance Analysis of Machine Learning Algorithms to Classify Software Requirements. *Journal of Electrical Systems (JES)*, 20(2), 1588–1599. <https://journal.esrgroups.org/jes/article/view/1464/1196>
- PROMISE Repository. *Public dataset of functional and non-functional software requirements*. University of Ottawa. <https://promise.site.uottawa.ca/SERepository/>

8. Appendix

I've uploaded the two documents to the shared folder as requested. Please let me know if you're unable to access them.

[Graduation Project](#)