



UNIVERSIDAD DE LAS AMÉRICAS PUEBLA

ENGINEERING SCHOOL

DEPARTMENT OF INFORMATICS, ELECTRONICS AND MECHATRONICS

Desarrollo de Aplicaciones Móviles (LIS4012)

LAB 07: Project Progress

Professor: Julio Noé Hernández Torres

Cecilia Soriano Tochimani, 173633

Melisa Sampieri Espinoza, 173948

Luis Felipe Jarquin Romero, 186812

Spring 2025

March 13th, 2025

Tasks Accomplished by Each Team Member by March 6th

Team member 1: Cecilia Soriano Tochimani

Cecilia has been focused on developing the app's user interface using SwiftUI in Xcode. She has designed and implemented various views, ensuring a seamless and intuitive user experience. Her work includes creating layouts, animations, and navigation flows to make the app visually appealing and user-friendly. Additionally, she has been refining the app's responsiveness and accessibility to ensure a high-quality user experience across different iOS devices.

Tasks done:

- The splash frame has been completed.
- The home view is in progress and has advanced to 30% completion.

Team member 2: Melisa Sampieri Espinoza

Melisa has been working on integrating the BERT pretrained model into the app for emotion recognition and tone analysis. She is responsible for adapting the model to fit the specific needs of the app, leveraging her experience with BERT. Additionally, she has been testing different hyperparameter configurations to optimize performance. She is also exploring ways to connect the model's output to the app's views for real-time suggestions while ensuring efficiency and accuracy.

Tasks done:

- Trained multiple versions of the BERT model to evaluate which performs best for our application.

Team member 3: Luis Felipe Jarquin Romero

Luis has been focused on integrating the OpenAI API to enhance the app's functionality, particularly for text rewriting and tone refinement. He is working on establishing a secure connection between the app and the API to ensure smooth data processing. Additionally, he is optimizing the API's performance to provide quick and accurate responses to user inputs. To achieve this, he has collaborated with another team member to research the best implementation strategies and ensure compliance with security standards. He has also been involved in approving and reviewing the progress made by other team members to ensure alignment with project goals.

Tasks done:

- Conducted research with a teammate on how to effectively implement the OpenAI API.
- Approved and reviewed the progress of different implementation aspects to ensure consistency and efficiency and responsible to make the documentation

Tasks Accomplished by Each Team Member by March 13th

Team member 1: Cecilia Soriano Tochimani

Cecilia has been focused on developing the app's user interface using SwiftUI in Xcode. She has designed and implemented various views, ensuring a seamless and intuitive user experience. Her work includes creating layouts, animations, and navigation flows to make the app visually appealing and user-friendly. Additionally, she developed another view for the app, keeping it clean and easy on the eyes.

Tasks done:

- Fixed errors in the home view.
- Implemented a feature where the identified sentiment is highlighted with color, fulfilling one of the app's requirements.

Team member 2: Melisa Sampieri Espinoza

Melisa worked on converting the BERT model into a CoreML so that it can be implemented in Swift UI.

Tasks done:

- Find versions of BERT to include it in the application
- https://huggingface.co/docs/transformers/model_doc/distilbert#tfdistilbertformaskedlm
- <https://apple.github.io/coremltools/docs-guides/source/convert-tensorflow-2-bert-transformer-models.html>
- <https://huggingface.co/dlaiymani/bert-base-ner-coreml>
- <https://github.com/huggingface/swift-coreml-transformers/tree/master>

Team member 3: Luis Felipe Jarquin Romero

Luis researched methods for cybersecurity. A log in is obviously necessary and there may be messages that the user would like to further protect. Also, it is important to encrypt users' passwords. After considering Argon2, PBKDF2, and bcrypt, we decided to go with bcrypt because of how easy it is to implement while maintaining strong security [1]. It is much faster than Argon2, as the hashing is done in less than one second [2].

Tasks done:

- Deciding on bcrypt for Swift UI called SwiftBcrypt.
- Updating documentation on GitHub.

References:

1. T. P. Batubara, S. Siburian, and R. T. A. Pasaribu, "Analysis Performance BCRYPT Algorithm to Improve Password Security from Brute Force," *J. Phys.: Conf. Ser.*, vol. 1811, no. 1, p. 012129, 2021. Available: <https://doi.org/10.1088/1742-6596/1811/1/012129>.
2. Stytc, "Argon2 vs. bcrypt vs. scrypt: Which password hashing algorithm should you use?" *Stytc Blog*, May 2022. Available: <https://stytc.com/blog/argon2-vs-bcrypt-vs-scrypt>.

GitHub link: <https://github.com/LuisAtMcGill/vibecheck.git>