PCMJ

Progress in Computer and Mathematics Journal (PCMJ)
volume 3 [November, 2025]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

GRAMMARCHECK: A DYNAMIC LL(1) GRAMMAR CHECKER USING RECURSIVE DESCENT PARSING WITH CHATBOT ASSISTANCE

NUR ATHIRAH ARAMI ROSELAN
nurathiraharamiroselan@gmail.com

NOOR HASIMAH IBRAHIM TEO*
shimateo@uitm.edu.my

Article Info

Received: March 2025
Accepted: September 2025
Available Online: November 2025

Abstract

This project focuses on improving the understanding and application of LL(1) grammar in compiler design. LL(1) grammar is crucial for parsing algorithms, but despite that, students often find it difficult to grasp, especially when it comes to applying the 12-step solution process. Current tools provide general solutions but lack detailed guidance. In order to address this, the project develops an Android mobile application chatbot that helps users understand and solve LL(1) grammar problems interactively. The application includes an automated checking feature to assist users in verifying their steps and answers. It provides solutions for valid LL(1) grammars through a structured, step-by-step process, offering feedback on correct, incomplete, or incorrect answers. Additionally, the chatbot was using a modified waterfall methodology and was implemented using recursive descent parsing, an essential method for understanding LL(1) grammar, by allowing users to input and test grammar rules. Testing and validation were performed on the application, where it showed great results, particularly in validation output testing, by producing the LL(1) grammar solution without fail. In all validation test cases, the application passed every single one, proving that the application can generate accurate solutions. Moreover, usability testing demonstrated that 75% of the users felt the app greatly reduced the complexity in the solution process of the 12 steps in LL(1) grammar, while 100% found the automated checking feature useful. Given feedback from the app was clear and well-received, the overall user satisfaction was high. The tool has the potential to significantly enhance learning for students studying LL(1) grammar and offer a practical solution for parsing challenges in compiler design. Future work could focus on improving performance and extending the system's use to other areas in computer science education.

Keywords: LL(1) Grammar, Chatbot, Grammar Checker, 12-Step Solution, Testing

Progress in Computer and Mathematics Journal (PCMJ)
volume 3 [November, 2025]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

INTRODUCTION

This section provided discussion regarding the proposed project including the background study, problem statements, the objectives and aim of the project, scope, and lastly the project significant.

Background Study

Online education has grown significantly, offering adaptability and accessibility, especially for those pursuing higher education (Lockee, 2021). The pandemic accelerated digital adoption, leading to Education 4.0, where online learning became essential (Boca, 2021). Furthermore, chatbots, using AI and machine learning, are increasingly used in Computer Science and Engineering education, including compiler design (Tamrakar & Wani, 2021). This project develops a dynamic chatbot to assist students in solving LL(1) grammar problems, which involve complex steps like computing FIRST and FOLLOW sets (Bergmann, 2017). Additionally, these steps are crucial in parsing but can be challenging for students due to the need for precise rule application, handling nullable rules, and tracking multiple dependencies within a grammar. Moreover, students must understand and remember the 12-step computation process, which adds to the difficulty (Dol, 2023). Traditional teaching struggled with abstract topics and delayed feedback (Jovanović et al., 2022), while online education improved accessibility (Cavalcanti et al., 2021). This results in a need for real-time assistance, where a dynamic chatbot enhances student learning and engagement (Riabko et al., 2023).

Problem Statement

Despite the shift to online learning, compiler theory students struggle with timely feedback. Traditional classrooms offer immediate instructor support, but online environments lack this, making it harder for students to correct mistakes (Erlangga, 2022). Instructors also face challenges in providing sufficient feedback, especially in large cohorts (Cavalcanti et al., 2021). Delayed feedback slows learning and reduces motivation, highlighting the need for tools that bridge traditional and online learning with real-time support.

While chatbots can enhance education through personalized assistance (Mathew, Rohini, & Paulose, 2021), many rely on retrieval-based methods, limiting their ability to address students' diverse needs (Aleedy et al., 2022). As a result, students may not receive the tailored

guidance necessary for mastering compiler theory, worsening the challenges of traditional teaching.

Objectives

This project aims to develop a LL(1) grammar checker to find the selection set of LL(1) type of grammar in the compiler subject by returning the output of 12 steps with chatbot assistance.

The objectives of this project are:

i. To design a dynamic grammar checker chatbot (GrammarChecker) for the 12-step LL(1) type of grammar in the compiler subject.
ii. To develop a GrammarChecker chatbot using a recursive descent parser.
iii. To test the functionality, validation and usability of the GrammarChecker.

Scope

The scopes of this project are:

a. Language

The language that will be used in this project development is English language.

b. Target Subject

Compiler subject but specifically covers the LL(1) grammar selection set.

c. Target User

Students and lecturers who learn compiler subject and need to find the selection set of LL(1) grammar by using 12 steps.

d. System Function

- LL(1) Grammar Solution Generation: To provide 12 steps LL(1) solution.
- LL(1) Grammar Checker: For checking answer provided by user.
- Feedback Generator: To give feedback to user whether the answer input by user is correct, incomplete or correct.

e. Type of Chatbot

This chatbot is dynamic chatbot where it receive dynamic LL(1) grammar input and generate the solution instantly instead of stored the LL(1) grammar problem and pre-defined answer in database.

Significance of Study

By developing the GrammarChecker mobile application, students gain 24/7 support and real-time feedback to better understand LL(1) grammar. This tool helps lecturers efficiently evaluate responses and provide timely guidance. With its flexibility, engagement, and instant assistance, the app enhances the learning experience, making compiler grammar more accessible and manageable.

USE CASE DIAGRAM

Figure 1 shows the interaction between the user of the system and the chatbot. From this figure, it can be identified what task the user 50 and the chatbot should or can do in this development of the mobile application.

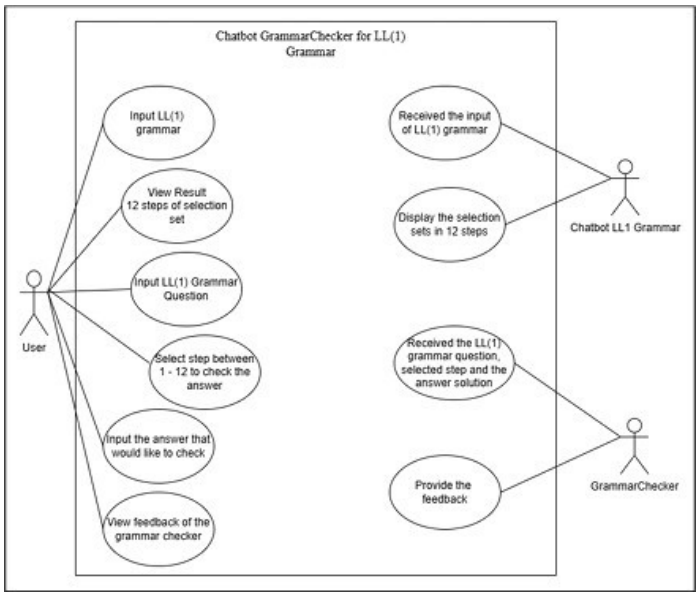

Figure 1: Use Case Diagram

INTERFACE DESIGN

This section provides an overview of the user interface design for the GrammarChecker mobile application. Each design page is explained to serve as a reference during the

implementation phase. Having a clear reference ensures consistency in the final design, aligns with user expectations, and enhances overall user experience.

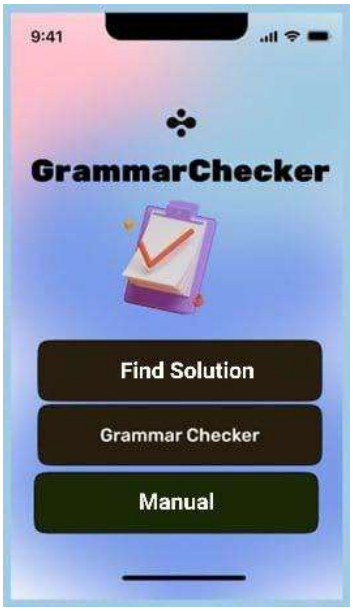

Figure 2: Main Page

Figure 2 is the main page for the chatbot checker where the user can choose to use the function for find solution of LL(1) grammar, grammar checker or to know more about the application.

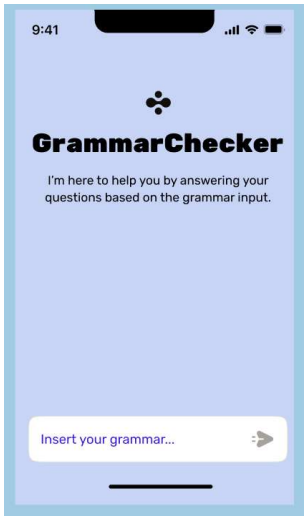

Figure 3: Input Grammar Interface

Figure 3 shows that this is the interface where the user can start to input their LL(1) grammar.

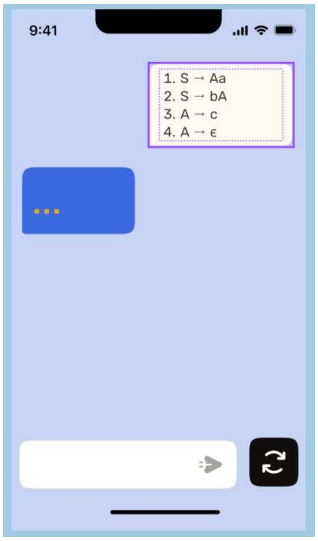

Figure 4: User Insert Input Interface

Following Figure 4, once the user inserts the LL(1) grammar input, the chatbot will immediately give their response which is the answer for all 12 steps to find the selection sets based on the input from the user.

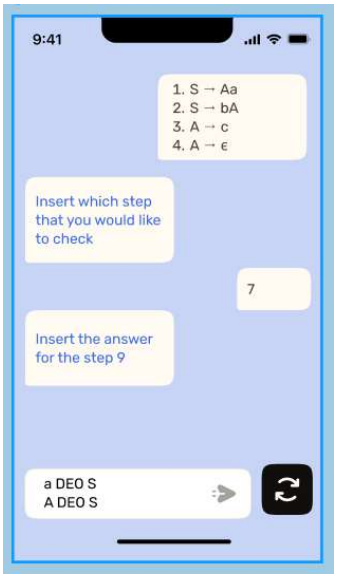

Figure 5: Grammar Checker Module Interface

Figure 5 shows that if the user chooses to click on module grammar checker, it will direct them to this page where the user needs to insert the LL(1) grammar, then choose which step and

insert the answer to be checked for that step before the chatbot can give a feedback response of your answer.

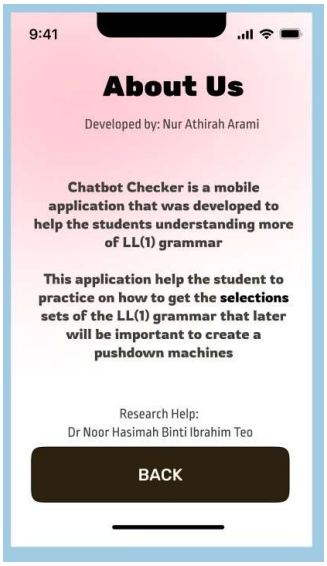

Figure 6: Manual Page Interface

This Figure 6 shows the details of the chatbot mobile application such as who is the developer of this mobile application and how to use the mobile application.

RESULT AND DISCUSSION

This part presents the testing and evaluation of the GrammarChecker mobile application, focusing on usability testing. These tests ensure the system meets its objectives by providing accurate solutions, validating user input, and offering appropriate feedback to the end user.

Usability Testing

Usability testing was conducted through a survey of 20 students who had taken the CSC569 Principles of Compiler course. The evaluation focused on ease of use, feedback clarity, navigation, and overall satisfaction. Results indicated that students found the app intuitive, effective in simplifying LL(1) grammar solutions, and beneficial for learning. The automated checking feature was particularly praised. Figure 4 illustrates the positive reception, with all respondents willing to recommend the app.

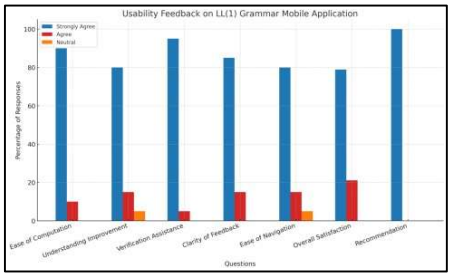

Figure 4: Usability Testing Result

CONCLUSION

The GrammarChecker mobile application successfully achieves its objectives and addresses the problem statement by providing accurate LL(1) grammar solutions. It passed functionality testing, ensuring efficient performance across various scenarios, while expert validation confirmed the correctness of its outputs. Additionally, usability testing received positive feedback, highlighting the app's reliability, ease of use, and effectiveness in assisting students with LL(1) grammar learning. These results demonstrate that the application is a valuable tool for improving understanding and problem-solving in LL(1) grammar.

REFERENCES

Aleedy, M., Atwell, E., & Meshoul, S. (2022). Using AI chatbots in education: recent advances challenges and use case. *Artificial Intelligence and Sustainable Computing: Proceedings of ICSISCET 2021*, 661-675.

Bergmann, S. D. (2017). Compiler design: Theory, tools, and examples (Version 2). Open Educational Resources. Retrieved from https://rdw.rowan.edu/oer/1/.

Boca, G. D. (2021). Factors influencing students' behavior and attitude towards online education during COVID-19. *Sustainability, 13*(13), 7469.

Cavalcanti, A. P., Barbosa, A., Carvalho, R., Freitas, F., Tsai, Y.-S., Gašević, D., & Mello, R. F. (2021). Automatic feedback in online learning environments: A systematic literature review. *Computers and Education: Artificial Intelligence, 2*, 100027.

Erlangga, D. T. (2022). Student Problems in Online Learning: Solutions to Keep Education Going on. *Journal of English Language Teaching and Learning, 3*(1), 21-26

Jovanović, N., Stamenković, S., Miljković, D., & Chakraborty, P. (2022). ComVIS—Interactive simulation environment for compiler learning. *Computer Applications in Engineering Education, 30*(1), 275-291.

Lockee, B. B. (2021). Online education in the post-COVID era. *Nature Electronics, 4*(1), 5-6.

Mathew, A. N., Rohini, V., & Paulose, J. (2021). NLP-based personal learning assistant for school education. Int. J. Electr. Comput. Eng, 11(5), 4522-4530.

Riabko, A. V., Vakaliuk, T. A., Zaika, O. V., Kukharchuk, R. P., & Kontsedailo, V. V. (2023). *Chatbot algorithm for solving physics problems.* Paper presented at the CEUR Workshop Proceedings.

Tamrakar, R., & Wani, N. (2018, April). Design and development of CHATBOT: A review. In *Proceedings of the International Conference on Latest Trends in Civil, Mechanical, and Electrical Engineering* (pp. 369–372). ResearchGate. https://www.researchgate.net/publication/351228837