UXNBot : An Approachable Way to Learn Basic Assembly Language with Low-carbon Computing

Yixuan Ruan

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

A dissertation presented in part fulfilment of the requirements of the Degree of Master of Science at The University of Glasgow

September 1st 2022

Abstract

Nowadays, smart devices are consuming a large portion energy, and software are getting more complex with functionalities that we'll probably never use, which take up many spaces and require a lot of computing resources. As a consequence, we have to buy new phones and computers to accommodate the change.

This project aims to build an **UXNBot** to make it easier and more insteresting for students and beginners to learn about UXN, a project that is established with the aim to utilise old devices with resource-aware computing, and the basic assembly language instructions it uses.

The **UXNBot** is build on one of the most popular Chinese social media platform **WeChat** and uses an interactive way to talk with the users, it is implemented using Express framework on Node.js environment.

This is particularly interesting as the Chatbot and UXN virtual machine and Uxntal Assembly language have never been used together on WeChat.

(Real evaluation data to be added afterwards)It was found that subjects using **UXNBot** achieved an average of 98.5 out of 100 which is much better than subjects reading materials tested an average score of 83.8, though it is less suitable for advanced learning purpose.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: ———————————— Signature: ————————————

Acknowledgements

Most of the figures and graphs in this dissertation are draw by myself with free icon downloaded from Flaticon.com.

The figures mentioned below are referenced.

Figure 2.4 and Figure 2.5 is a direct copy of the images of EJAVA-Bot[11].

Figure 2.7 Python-bot Usage Examples, which is direct copy of Python-bot Running Example on SnatchBot platform by Okonkwo and Ade-Ibijola[31].

Figure 5.1 Learning Module Flow Chart contains images copied from XXIIVV.com, which are licensed under https://creativecommons.org/licenses/by-nc-sa/4.0/.

Contents

Chapter 1: Introduction

This chapter firstly introduces the motive and context behind this project, answering why UXNBot - the assembly language teaching Chatbot, is developed. Then the basic work of the project is briefly mentioned, with a last section giving an overview of the dissertation structure.

1.1 Motive & Context

1.1.1 Motive

UXN is a project initiated toward sustainable computing to run small program suitable for old devices and it uses Uxntal assembly language. This project is meant to build a Chatbot to narrow the gap between beginners and Uxntal assembly language, and makes people aware of the concept of low-carbon or sustainable computing.

Why Sustainable Computing?

Around 2% of global CO_2 emission and 10% of global energy consumption are caused by IT devices in 2007 [9]. With the advancement of cloud computing, data centers alone will cause 1/5 of world's power supply by 2025 [10]. Another phenomenon that should bring focus is old machines and devices which only require low level of electricity are dumped because of CPU and memory demanding softwares. Find a way to utilise those obsolete machines is a way to a more sustainable future.

Why Learning Assembly Language?

High level languages usually take more memory and CPU to run than low level languages, which makes ANSI C and assembly languages great choices to use when developing softwares and systems on embedded or obsolete machines, they are also useful in kernel development and optimization part of compiler.

Although assembly language seem not to be used by most programmers, they are ranked 9th by TIOBE Index according to [36]. Assembly languages are certainly an important component in current industry.

What's more, learning assembly language is a perfect way to deepen our understanding of computer architecture, a solid computer science education process shouldn't be lack of learning some assembly language.

Why Uxntal & UXN ?

Not like other assembly languages, Uxntal is a much more simpler one with only 32 instructions, yet it possesses very complete abilities to build games, electrical instruments and various range of applications on UXN virtual machine, which is a very light weight one writen in ANSI C yet possesses the full power to run applications like games and instruments.

It's succinctness and power make it possible to run on very old machines, and its simplicity also provides strong educational capabilities.

Why Chatbot ?

If learning Python is a joy, then learning Java & C/C++ may not be a great experience. But when it comes to learning assembly languages, it is almost a daunting experience.

ChatBot is thought to be a tool that act like a toy and a friend[7], which is perfectly suitable to further bring close the gap between students and assembly language.

This project builds a ChatBot on the major social media WeChat, enables the students to talk with the UXN virtual machine and get immediate feedback on their input instructions without sitting in front a computer and setting up an development environment or install any more apps.

1.1.2 Context & Innovation

By investigating related works about using Chatbot on social media platform to instantly run instructions in an interpreter fashion and provide results and grammar feedback, no literature or projects published are found.

This project may be the first social media Chatbot to instantly run assembly language instructions and give instant feedback.

Some other related work about educational Chatbot and Uxntal & UXN project can be found on literature review chapter.

1.2 Main Work

1.3 Outline

Chapter 2:   Survey

2.1   Sustainable or Low Carbon Computing

Sustainable computing is a huge and comprehensive research direction. Every work that can save energy or reduce CO_2 emission for IT services may considered a part of it. We can get a large image by looking from several directions.

The energy that IT services consumes comprise cloud computing, fog computing and edge computing. Cloud computing means the services run on data centers, fog computing represents the computing takes place at local area network (LAN) devices like computers own by organizations to deal with computing rather than using the public cloud service and edge computing means the computing is happening on mobile devices, desktop computers and internet of things (IoT) devices like cpu inside refrigerator, air-conditioner etc.

For sustainable cloud computing, large data centers are transforming from thermal power to renewable energy source. Google started to purchase wind power from local developers where their data centers located since 2010, and they are determined to achieve their goal of 24/7 carbon-free energy (CFE) by 2030[27]. Despite the use of renewable energy, data centers are also struggling to improve their energy efficiency from different directions. We can get a glimpse of the portion of different sources of energy consumption in data centers[15] in Figure 2.1.
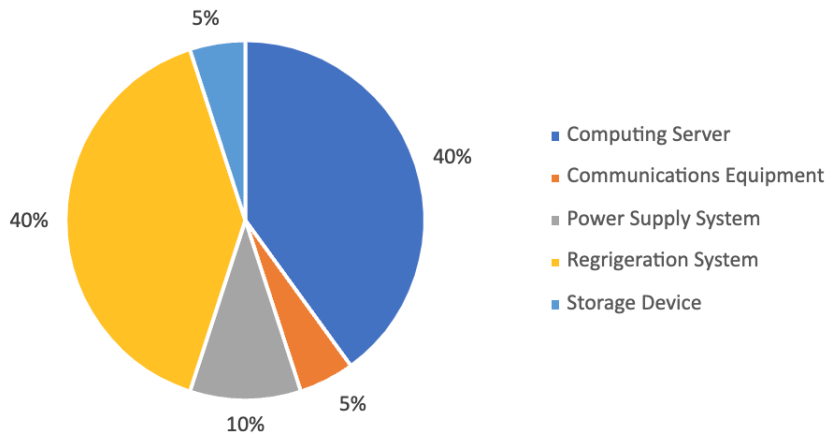

Figure 2.1: Data Centers Power Consumption Distribution

For sustainable fog and edge computing, it is important both to increase CPU efficiency [29] and to utilise cheap and old chips and devices to reduce waste, and that's where UXN come to help.

2.2   UXN

UXN virtual machine [22] is originally developed by Devine Lu Linvega using ANSI C. It can be run on MacOS, Linux and Windows with very easy setup process. It's emulator and program only takes several kilobytes to run without taking too much memory and CPU

power. But it is capable to do a lot of jobs like calculator, electric piano etc. Compare to software nowadays, which require hundreds of megabytes and even gigabytes, it is a perfect choice for sustainable computing.

2.2.1 Multiple Language Versions of Uxn Emulator

Since then, UXN virtual machine has been ported to many other languages by the community. Ismael-VC [14] write the Julia version of UXN, DeltaF1 [13] write the UXN using Lua. Thanks to max22's [28] work, we can use python to compile Uxntal and run UXN program. Devine Lu Linvega [23] also made it possible to use JavaScript to run UXN rom. These efforts contribute a lot to make it possible for UXN to work with other languages, which is a great step forward to enlarge its popularity.

2.2.2 Multiple System Adaptions of Uxn Emulator

As UXN project takes sustainable computing as its initiative, many passionate people have been porting UXN to other systems. Asiekierka [3] port UXN to Nintendo DS. Ivodopiviz are working on porting UXN to PlayStation Vita. Andrew Alderwick [2] are building UXN on the Raspberry Pi Pico. For mobile phone users, there are also projects undertaking. Kylestew [17] are building UXN emulator on iOS, James [33] are porting UXN on Nokia N9000. UXN can also be run on DOS system through Neauoire's work[30].

All these different language versions of UXN emulator and different hardware system adaptions that can run UXN emulator are building a strong infrastructure for developers to easily engage with UXN. Though it is still not approachable in every direction. It is a fast growing community.

2.3 Uxntal

If UXN virtual machine is the core of UXN project, then Uxntal programming language is the gate the attracts developers. The community has done many works to support developing UXN programs using uxntal language.

2.3.1 Uxntal Editor Support

Devine Lu Linvega are working on an Uxntal Editor [21] which support syntax highlight/navigation, proportional text, directory listing and a lot more.

There is also an option to code Uxntal online at LearnUXN [20].

If the developers want to code within their favorite editors, Uxntal language support is available in them as well. Karol Belina [5] built a plugin to support coding Uxntal in Visual Studio Code. For our favorite command line editor - Vim, Karol Belina [6] developed a vim plugin to support Uxntal syntax highlighting in command line editing mode. There are plenty of plugins for other editors like Nano[8], Atom [4] and Sublime [25] etc. Developers have plenty of choices to improve their development experience with Uxntal.

2.3.2 Uxntal Compilation Support

To compile Uxntal program to something that can be run by an UXN Virtual machine/Emulator, we also need compilers.

For compiler, the most important part is assembler, which convert Uxntal assembly language to byte code that UXN emulators can run. The original UXN assembler [22] is developed

by Devine using ANSI C. There is also a JavaScript version of UXN assembler written by Rafapaezbas [34]. Drifblim, which is also [19] written by Devine Lu Linvega is a Uxntal version of UXN assembler.

Grammar and syntax checker is also an important component. Devine Lu Linvega [24] implemented a linter providing warnings like redundancy, static-arithmetic, static-duplicate etc. This project is very useful for programmers to find bugs in Uxntal programs, and it is actively developing and maintaining.

We can even have disassemble a UXN rom, the UXN byte code program, using DeltaF1's PyUxntalDisasm [12] program.

2.4 UXN Educational Projects

Focusing on the educational part of UXN community, we can find many tutorials about UXN and Uxntal.

The first page that UXN beginners should read is the home page of the UXN project [18], it has detailed introduction to run a demo UXN program in almost all desktop computer systems.

Xxiivv.com [26] is also a good place to get to know Uxntal assembly language, where the stack design, opcode of Uxntal, and programming grammar are illustrated. Although xxiivv.com is a comprehensive site towards Uxntal, it may be hard to comprehend. An alternative way to learn Uxntal easily is through aN introduction to UXN online book, which is available at compudanzas.net [35] and it provides a progressing learning experience.

2.5 Chatbot

The surge of AI in the Natural Language Processing field has made Chatbot really welcoming these years because its excellent performance to improve the dialogue experience.

As we normally chat both through voice and text, we can also "chat" to machines through voice and text.

The general text Chatbot pipeline is pretty much the same as in Figure 2.2. We start by typing in text in the user interface, and the system will analyse our intent and match one of the predefined patterns, the following step is return a predefined answer or to extract parameters from the matched intent and process it, possibly including interacting with some data sources [1].

The other type of Chatbot - voice assistant, like Siri, Alexa and Cortana etc, is very similar to text Chatbot but with two additional modules[16]. The first one is a preprocessing module - Automatic Speech Recognition (ASR). It is used to clean the background noise, segment the waveform into words, and then transform those segments into words and sentences. The procedure follows is the standard text Chatbot processing pipeline. Another additional module included is to generate human like voice from results. The two additional modules are show in Figure 2.3 .

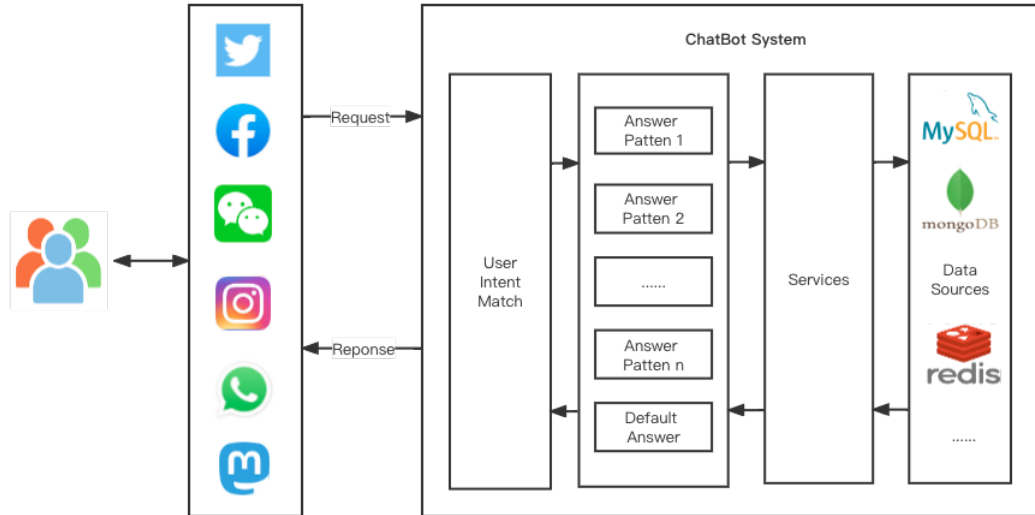We'll be focusing on text Chatbot in this project.
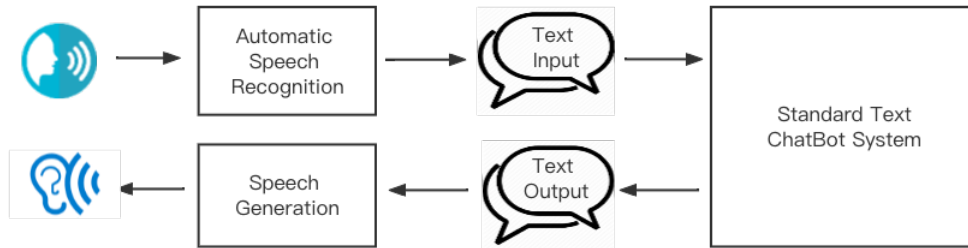
Figure 2.2: General Architecture of a Text Chatbot

Figure 2.3: Additional Modules for Voice Assistant

2.6 Educational Programming Language Chatbot

The recent research projects relating to educational Chatbot are mostly text Chatbot [32].

Looking specifically into Chatbot for teaching programming language, there are not two many researches available towards this direction. We will be looking at two Chatbot helping users to learn programming language: E-JAVA Chatbot and Python-bot.

2.6.1 E-JAVA Chatbot

Daud, Teo and Zain[11] developed a e-JAVA Chatbot comprise a concept learning and greeting module and a code generation module to generate selection and repetition code statements.

The concept and greeting module can respond to user saying hi to bot and asking about basic concept about JAVA programming, several test cases are shown in Figure 2.4.

The other module is Table2Code module shown in Figure 2.5, where the Chatbot generates code depending on how the user fill in the form.

Although the idea of generating selection and loop statements using a table or form that the user fills is quite innovative, it does seem to be over-complicated to learn about "form grammar" first. It should be better to provide syntax analysis/grammar checking to check the

statements user wrote rather than using table to code generation to make the JAVA learning curve even harder.

Test Data #5	User input: Hello!!!
Expected Result	Random: Greetings messages such as hi! Any questions about java control structures
Actual Result	Same as expected result
Remarks	Passed
Test Data #6	User message: Question related about java control structures Example: What is control structures?
Expected Result	Related answer corresponds to the questions
Actual Result	Same as expected result
Remarks	Passed
Test Data #7	User message: Question not related about java control structures Example: What is your name?
Expected Result	Error respond message corresponds to user input Example: Sorry, I don't think that can get your questions.
Actual Result	Same as expected result
Remarks	Passed

Figure 2.4: E-JAVA Chatbot Greeting and Concept Module[11]

Figure 2.5: E-JAVA Chatbot Table2Code Generation Module[11]

2.6.2 Python-bot

Chinedu and Abejid developed a Python-bot to assist Python teaching, there is not Python teaching Chatbot built before them [31].

The Python-bot is built using SnatchBot platform, which is a website for creating Chatbot. SnatchBot can be integrated with Facebook, Twitter, Skype and so on.

It has three modules as shown in Figure 2.6, including a concept learning module, a meeting scheduler module, a programming problem solving module.

The concept learning module is capable of providing concept of Data Types, Expressions, Control Statement, Strings and a lot more.

The meeting scheduler module can analyse the user input and automatically book a meeting between the user and a tutor or teacher through email.

Users can also try to solve some pre-defined programming problem by using Python-bot.

The actual interface and usage examples are shown in figure 2.7. A survey conducted at the University of Johannesburg, South Africa has shown that 73.7% of the students believed Python-bot is easy to use, 15.6% found it difficult and 10.7% were neutral, which is a good evidence for proving Python-bot is a user-friendly tool to use.
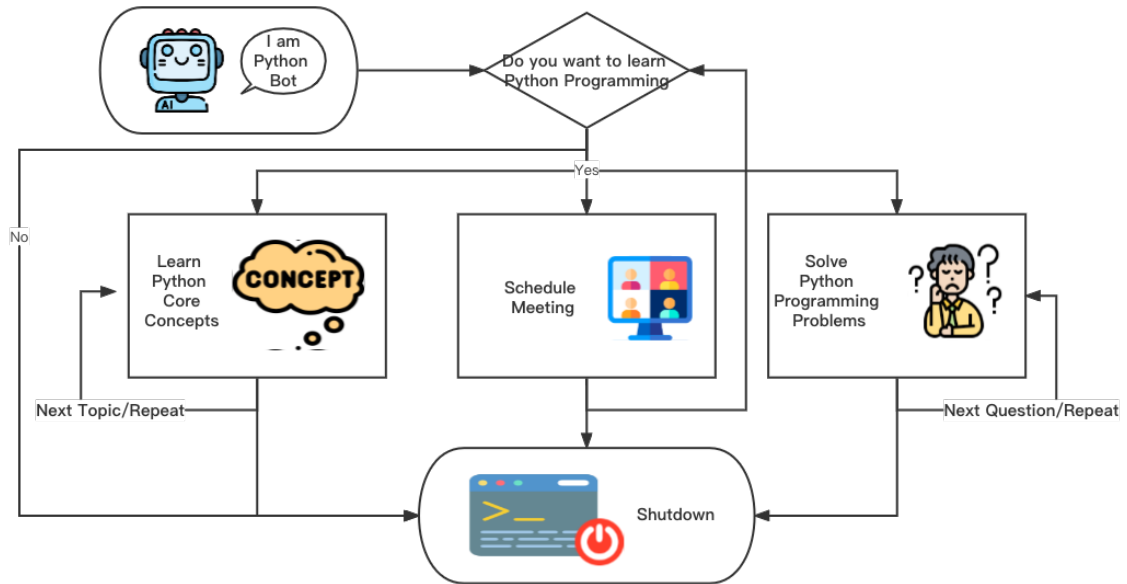

Figure 2.6: Python-bot Flow Chart

(a) Start Conversation   (b) Learn Python by Code Example   (c) Book Meeting with Tutor/Lecture
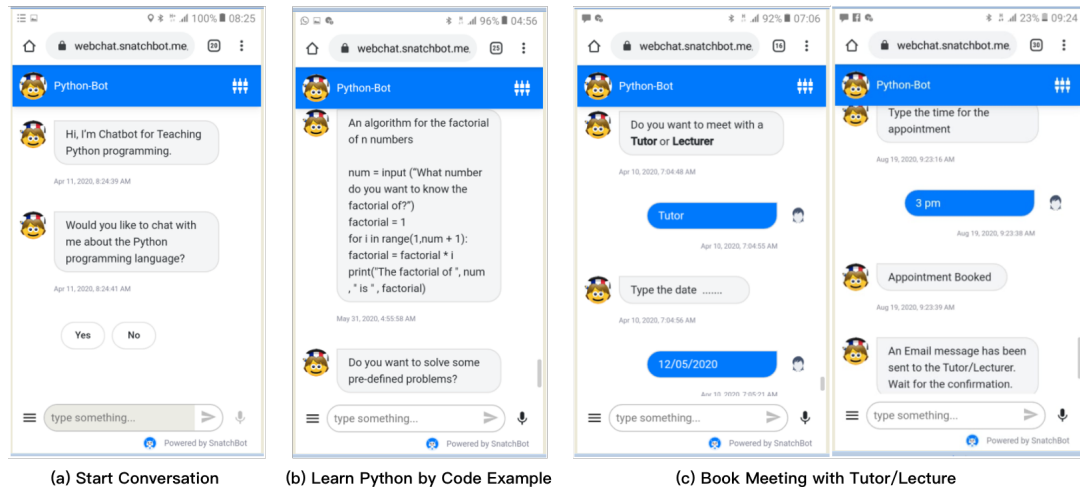
Figure 2.7: Python-bot Usage Examples[31]

E-JAVA Bot has its own innovation, Python-bot is a very easy-to-use Python learning Chatbot, but they all lack the ability to instantly run code or check grammar within the chat with the bot. What's more, they are both high-level programming language learning Chatbot. UXNBot is desgined to fill in the gap, and convey the concept of sustainable computing.

Chapter 3: Requirements
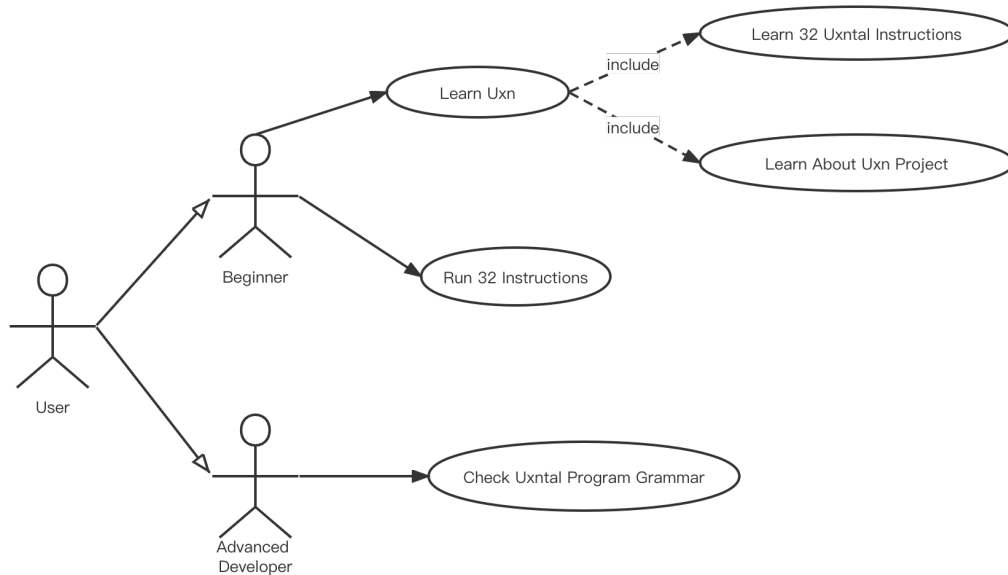
3.1 Functional Requirements

Figure 3.1: UxnBot Use Case Diagram

3.2 Non Functional Requirements

3.2.1 Usability

The target user is assembly language beginners, but possess some familiarity with high level programming language. The user need to know how to interact with the Chatbot easily.

3.2.2 Maintainability

As UXN community is still growing. The project should be well documented to allow others to do further development.

3.2.3 Performance Requirements

As a social media Chatbot intended to narrow the gap between beginners and Uxntal, it is important that we have a high availability, that means the users can interact with the Chatbot whenever they want and get a response. Due to limited number of target users, this project does not require high concurrency.
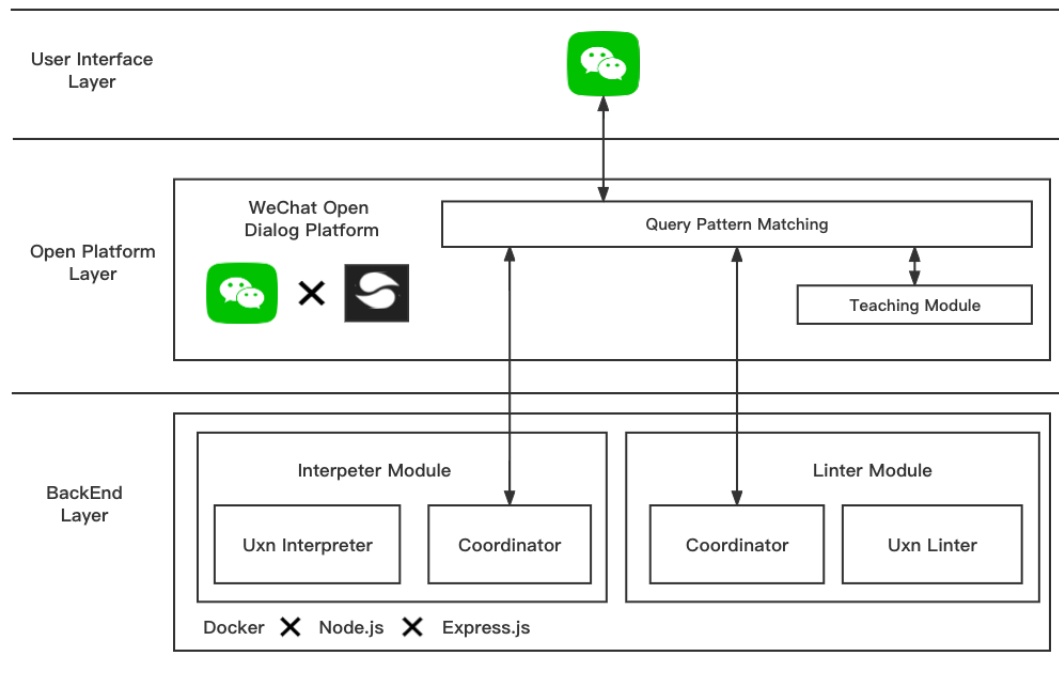
Chapter 4:   Design

Figure 4.1: UxnBot Architecture Design Diagram

4.1   Functional Module Design

4.1.1   Teaching Module

The teaching module is meant to give users information about UXN project, and teach students about how to

4.1.2   Linter Module

4.1.3   Interpreter Module

4.2   Technology Architecture Design

4.2.1   UXN Virtual Machine

4.2.2   WeChat Open Dialog Platform

4.2.3   Node.js & Express.js

4.2.4   Docker

Chapter 5: Implementation

5.1 Teaching Module

Figure 5.1: UxnBot WeChat Open Dialog Platform Teaching Module Implementation

5.2 Linter Module
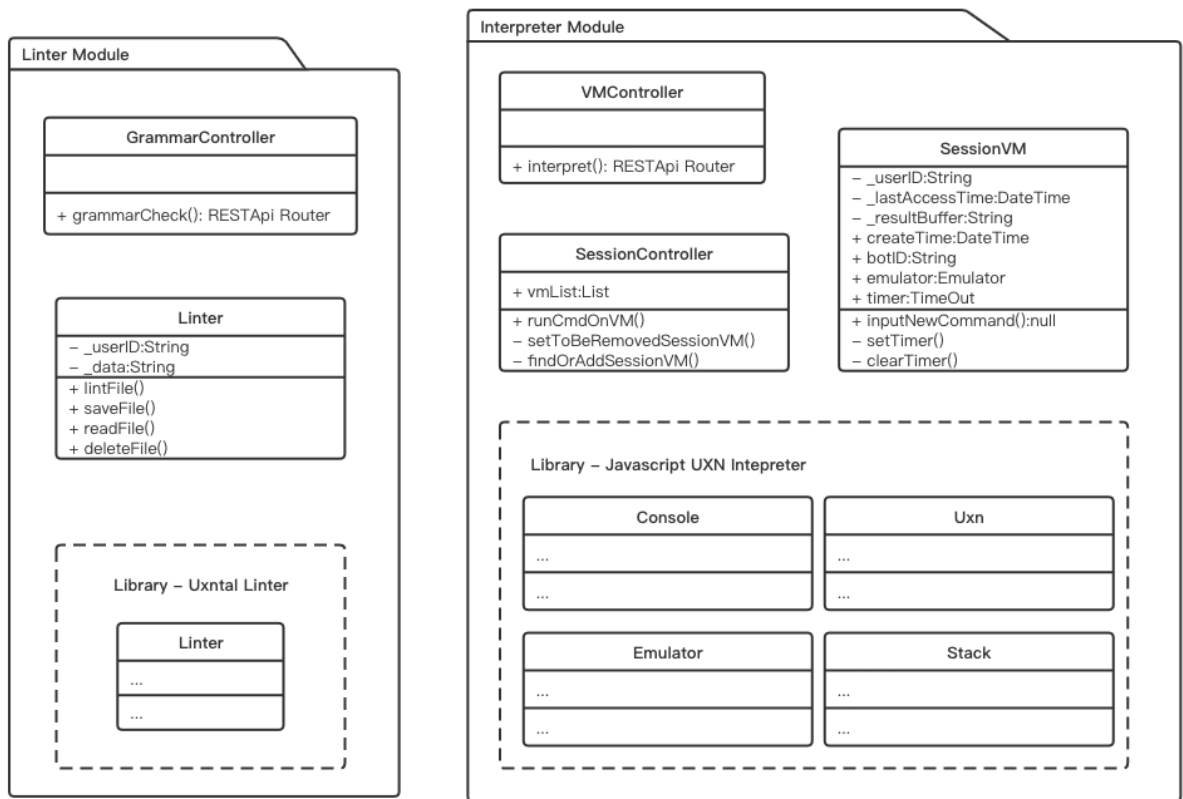
Sequence Diagram or Flow Chart for Teaching Module

Figure 5.2: Linter & Interpret Module Package & Class Diagram

5.3 Interpreter Module

Sequence Diagram or Flow Chart for Interpreter Module

5.4 User Interface

Sequence Diagram or Flow Chart for Interpreter Module

5.5 Deployment

The teaching module is built on WeChat open dialog platform. The interpreter module and linter module are built

Table 5.1: Deployment

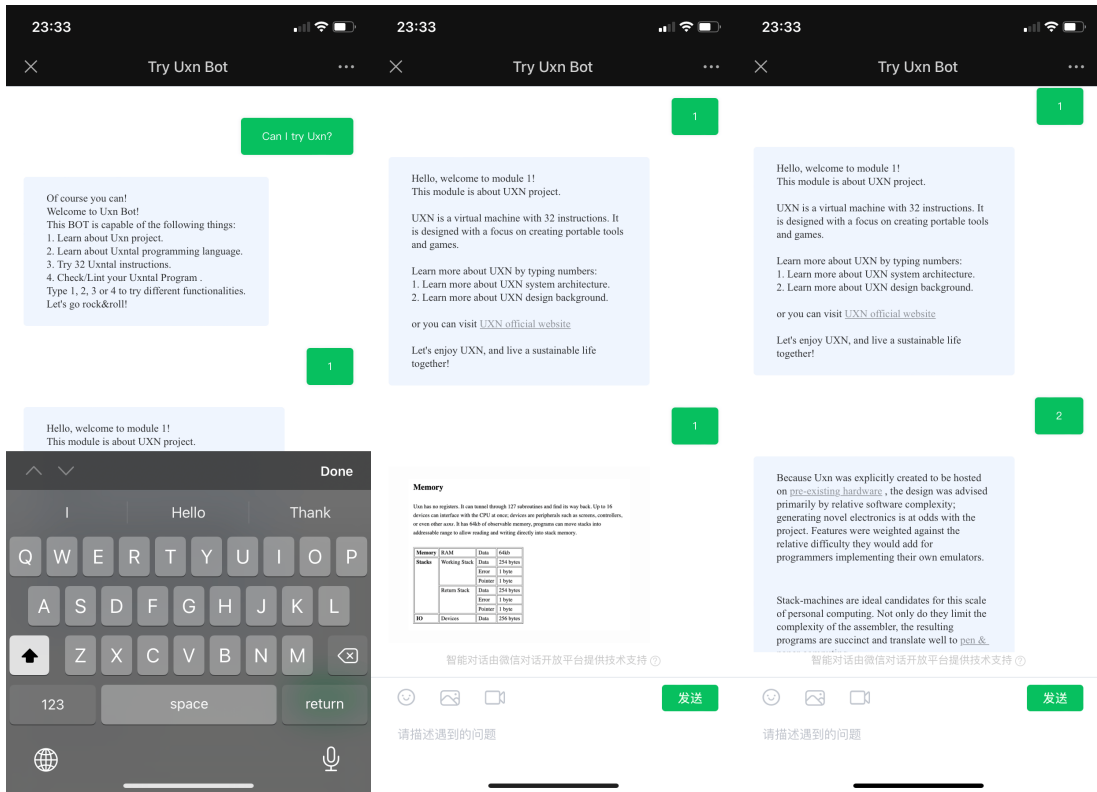| Module	Deployment Environment
Teaching Module	WeChat Open Dialog Platform
Linter Module	OVHCloud CentOS 7 x Docker
Interpreter Module	1Core + 2G RAM + 40G SSD

Figure 5.3: Teaching Module - Welcome Learning UXN User Interface

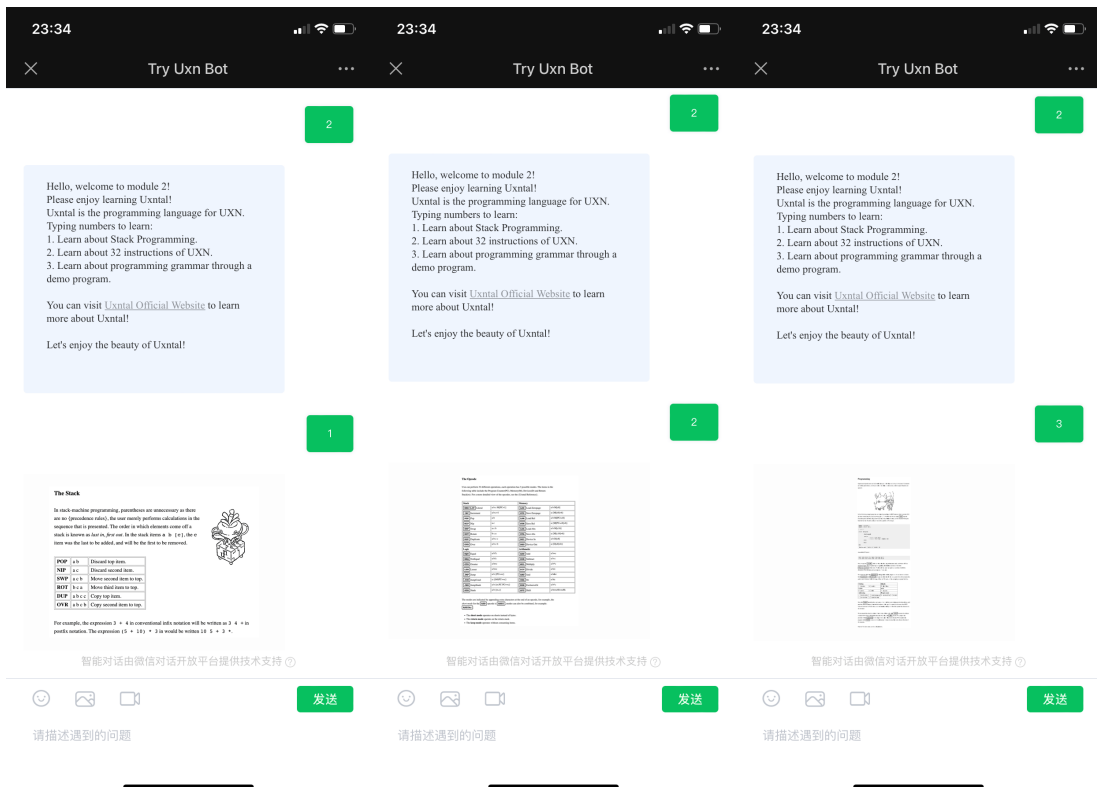Figure 5.4: Teaching Module - Learning Uxntal User Interface

Chapter 6:   Evaluation

Table 6.1: standard table

1	2	3	4
0.1	0.2	0.3	0.4

Chapter 7:   Conclusion

7.1   Summary

7.2   Reflection

7.3   Future work

Bibliography

[1] Eleni Adamopoulou and Lefteris Moussiades. An overview of chatbot technoloy. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 373–383. Springer, 2020.

[2] Andrew Alderwick. Pico uxn. `https://git.sr.ht/~alderwick/pico-uxn`, Nov 2021.

[3] Asiekierka. Uxnds: Nds port of the uxn virtual machine. `https://github.com/asiekierka/uxnds`, Feb 2022.

[4] Autumnull. Language-uxntal: Atom language support for uxntal, the programming language for the uxn virtual machine. `https://github.com/autumnull/language-uxntal`, April 2022.

[5] Karol Belina. Uxntal-vscode: Uxntal support for visual studio code. `https://github.com/karolbelina/uxntal-vscode`, Dec 2021.

[6] Karol Belina. Uxntal.vim: Uxntal support for vim. `https://github.com/karolbelina/uxntal.vim`, Dec 2021.

[7] Petter Bae Brandtzaeg and Asbjørn Følstad. Why people use chatbots. In *International conference on internet science*, pages 377–392. Springer, 2017.

[8] Cassvs. ~cassvs/uxntal-nano - sourcehut git. `https://git.sr.ht/~cassvs/uxntal-nano`, May 2022.

[9] Wikipedia Contributors. It energy management. `https://en.wikipedia.org/wiki/IT_energy_management`, Apr 2022. Last accessed: 2 Aug 2022.

[10] Danfoss.com. Data center power consumption. `https://www.danfoss.com/en/about-danfoss/insights-for-tomorrow/integrated-energy-systems/data-center-power-consumption/`, 2022. Last accessed: 2 Aug 2022.

[11] Siti Hawa Mad Daud, Noor Hasimah Ibrahim Teo, and Nurul Hidayah Mat Zain. Ejava chatbot for learning programming language: A post-pandemic alternative virtual tutor. *International Journal*, 8(7):3290–3298, 2020.

[12] DeltaF1. Pyuxntaldisasm. `https://github.com/DeltaF1/pyuxntaldisasm`, Dec 2021.

[13] DeltaF1. Deltaf1/uxn-lua: Lua + love2d port of the uxn virtual machine. `https://github.com/DeltaF1/uxn-lua`, Jan 2022.

[14] Ismael-VC. Uxn.jl: An assembler and emulator for the varvara/uxn stack-machine ported to julia. `https://github.com/Ismael-VC/Uxn.jl`, Dec 2021.

[15] Peter Johnson and Tony Marker. Data centre energy efficiency product profile. *Pitt & Sherry, report to equipment energy efficiency committee (E3) of The Australian Government Department of the Environment, Water, Heritage and the Arts (DEWHA)*, 2009.

[16] Veton Kepuska and Gamal Bohouta. Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home). In *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*, pages 99–103. IEEE, 2018.

[17] Kylestew. Uxnios: A uxnemu for ios. `https://github.com/kylestew/UxniOS`, July 2021.

[18] Devine Lu Linvega. 100r — uxn. `https://100r.co/site/uxn.html`, May 2022.

[19] Devine Lu Linvega. Drifblim: a uxntal assembler written in uxntal. `https://git.sr.ht/~rabbits/drifblim`, Aug 2022.

[20] Devine Lu Linvega. Learn uxn. `https://metasyn.srht.site/learn-uxn/`, Jul 2022.

[21] Devine Lu Linvega. Left: an uxntal text editor. `https://git.sr.ht/~rabbits/left`, Aug 2022.

[22] Devine Lu Linvega. Uxn. `https://git.sr.ht/~rabbits/uxn`, Aug 2022.

[23] Devine Lu Linvega. Uxn5: An emulator for the uxn stack-machine written in javascript. `https://git.sr.ht/~rabbits/uxn5`, 2022.

[24] Devine Lu Linvega. Uxnlin: A linter written in uxntal for the uxntal programming language. `https://git.sr.ht/~rabbits/uxnlin`, Aug 2022.

[25] Devine Lu Linvega. Uxntal sublime synta. `https://git.sr.ht/~rabbits/uxn/tree/main/item/etc/syntax-highlight/tal.sublime-syntax`, Aug 2022.

[26] Devine Lu Linvega. Xxiivv — uxntal. `https://wiki.xxiivv.com/site/uxntal.html`, Aug 2022.

[27] Oman Marc and Fernandez Stearns Ignacio. Clean energy projects begin to power google data centers. `https://cloud.google.com/blog/topics/sustainability/clean-energy-projects-begin-to-power-google-data-centers`, April 2022. (Accessed on 08/18/2022).

[28] Max22. Uxnemu.py: A uxn emulator written in python. `https://github.com/max22-/uxnemu.py`, Apr 2022.

[29] Jozef Mocnej, Martin Miškuf, Peter Papcun, and Iveta Zolotová. Impact of edge computing paradigm on energy consumption in iot. *IFAC-PapersOnLine*, 51(6):162–167, 2018.

[30] Neauoire. Uxn vga: The dos/vga uxn emulator. `https://git.sr.ht/~rabbits/uxn-vga`, Feb 2022.

[31] Chinedu Wilfred Okonkwo and Abejide Ade-Ibijola. Python-bot: A chatbot for teaching python programming. *Engineering Letters*, 29(1), 2020.

[32] Chinedu Wilfred Okonkwo and Abejide Ade-Ibijola. Chatbots applications in education: A systematic review. *Computers and Education: Artificial Intelligence*, 2:100033, 2021.

[33] James Palmer. Uxn900. `https://git.sr.ht/~foura/uxn900`, Aug 2022.

[34] Rpaezbas. Uxnasm-js: Javascript uxntal assembler. `https://github.com/rafapaezbas/uxnasm-js`, Dec 2021.

[35] Sejo. compudanzas — uxn tutorial. `https://compudanzas.net/uxn_tutorial.html`, Jun 2022.

[36] StatisticTimes. Top computer languages. `https://statisticstimes.com/tech/top-computer-languages.php`, 2022. Last accessed: 4 July 2022.

Appendix A: First appendix

A.1 Section of first appendix

Appendix B: Second appendix