# Stamford Chat Project Documentation

Project Title: Stamford Chatbot

Course Code: ITE222

Course Name: Programming II

Student (1) Name: Lynn Thant

Student (1) ID: 2406120001

Student (2) Name: Zwe Pyi Phyo

Student (2) ID: 2406120005

Course: ITE:222

Instructor: Dr Nay Myo Sandar

Date:

Table of Contents

[Introduction 3](#_Toc198548756)

[Objectives 3](#_Toc198548757)

[Tools and Technologies Used 3](#_Toc198548758)

[System Overview 3](#_Toc198548759)

[Class Descriptions 3](#_Toc198548760)

[Core Concepts Demonstrated 3](#_Toc198548761)

[Flowchart 4](#_Toc198548762)

[Algorithm 4](#_Toc198548763)

[Screenshots / Sample Outputs 4](#_Toc198548764)

[Conclusion 4](#_Toc198548765)

[11. References 4](#_Toc198548766)

# Introduction

In today’s digitally-driven educational landscape, universities are increasingly adopting chatbot technology—particularly using robust programming languages like Java to enhance student engagement helping many students to complete their respective tasks. These Java-based chatbots are being powered by the powerful programming language known as Java. From handling interactions like undergraduate inquiries and academic guidance to assisting with academic planners like GPA and budgets calculators. This type programming language had enhanced t development of secure, scalable, and cross-platform chatbot systems. In this developmental project, the Java-based chatbot not only improve communication between students and university services but also reduce the administrative burden on staff. Many developers believed that Java stands out as a powerful tool for building intelligent and reliable chatbot solutions tailored to academic environments.

Many universities considered using chatbot especially in their website. On the other hand, many developers suggested that implementing a chatbot in a website might be challenging due to High Initial Developmental Effort. That is where the chatbot application came into use. These type of chatbot applications uses Java programming language to simplify their developmental complexity. Instead of implementing on the website which depends mainly on internets and hosting Java application allows the users to run offline. Last but not least, these types of application provide more secure protocol than the web-based chatbots as it was being less exposed to the public due to proper local data handling.

# Objectives

In this project, the developers decided to implement features to support both students and lectures to be able to simplify their daily basis. The primary objective of this project is to develop a Java-based chatbot system that automates student support services within a university setting. The chatbot aims to provide instant, 24/7 assistance for common student inquiries related to admissions, course registration, examination schedules, fees, and general campus information. After careful consideration, the team has decided to go with these features for this chatbot.

|  |  |  |
| --- | --- | --- |
| Features | Functions | Advantages of using the features |
| GPA\_Calculator.java | CalculateGPA() | Helps students to calculate GPA based on their mark inputs. |
| Student\_Acc\_Create.java | Create\_Account();  CheckDetail(); | Helps students to create accounts based on their confidential inputs. |
| Student\_Budget.java | BudgetCalculator(); | Helps students to calculate budget based on their mark inputs. |
| Academic\_Calendar\_Exam.java | Check\_ Exam (); | Helps students to check exam date based on their majors. |
| Academic\_Calendar\_Holiday.java | Check Holiday(); | Helps students to check holidays for this semester. |

# Tools and Technologies Used

|  |  |
| --- | --- |
| **Category** | **Tools / Technologies** |
| **Programming Language** | Java (JDK) – for core development and execution |
| **Development Environment** | Eclipse IDE – for writing, debugging, and managing Java code. |
| **Execution Interface** | Command Line / Terminal – for compiling and running the application |
| **Java Packages Used** | -Scanner– for reading user input  - Maths – for calculations (e.g., GPA, budgets)  - Random– for generating simulated responses or data  -JOptionPane – for creating simple graphical dialogs. |

# System Overview

This **Java-based university chatbot system** is a standalone desktop application purpose-built to streamline communication between students and university services while enhancing academic support in an efficient and secure manner. Developed entirely using the **Java programming language**, the system capitalizes on Java’s platform independence, strong security features, and robust object-oriented capabilities to deliver a responsive and user-friendly experience without relying on internet connectivity or web hosting infrastructure.Designed with a **student-first approach**, the chatbot engages users through either a **console-based interface** using Java’s Scanner class or a **graphical user interface** using JOptionPane. This flexibility ensures that the system can be deployed in various settings, including computer labs, academic offices, or personal devices, regardless of network availability. The absence of external server dependencies also enhances **system reliability**, making it ideal for environments where internet access is limited or intermittent.

All interactions occur in **real-time**, allowing users to receive instant feedback and guidance. **It also provides tools** such as a **GPA calculator**, which helps students evaluate their academic performance, and a **budget planner**, which aids in managing academic-related expenses. The underlying logic is handled through structured Java classes and conditional logic, while mathematical computations (for GPA or budgeting) leverage the Math package. Randomized responses can be generated via the Random class to make the chatbot feel more dynamic and human-like.

In addition to improving the student experience, this chatbot system also alleviates pressure on university administrative staff by automating responses to routine queries and providing reliable, around-the-clock academic support. It represents a cost-effective and scalable solution for institutions seeking to **modernize their student services without the complexity and overhead of web-based deployment**.In summary, this Java-based chatbot system is a secure, efficient, and intelligent solution that aligns with the growing need for digital tools in higher education. It delivers meaningful support to students while demonstrating the versatility and strength of Java in academic software development.

# Class Descriptions

List and explain each class used in the project, including its responsibilities and methods.

# Core Concepts Demonstrated

Explain how the following were used in your project:

* Loops (for, while, do-while, if, nested if, switch)
* Constructors and Method Overloading
* Inheritance (classes and subclasses)
* Object Instantiation
* Use of 'this' and dot operator
* Java Packages (Scanner, Math, Random, JOptionPane)
* Encapsulation (getters/setters, private fields)
* Object interaction
* 2D Arrays
* Static and non-static methods
* Commenting and proper naming
* Access Specifiers (public, private, protected)

# Flowchart

Insert the flowchart for the whole project, including the classes.

# Algorithm

|  |  |  |
| --- | --- | --- |
| **Step (1.0)** | | **Objective: Welcoming User and Creating an Accounts** |
| **Algorithm** | **Procedure** | **Output (Screen Shot)** |
| 1.0 | * Display: “Welcome to Stamford Academic Help Chatbot!”. * Allow the user to choose whether they are students or lecturer. * Let the user to create an account to use the chatbot. |  |

|  |  |  |
| --- | --- | --- |
| **Step (2.0)** | | **Objective: Present User Options** |
| **Algorithm** | **Procedure** | **Output (Screen Shot)** |
| 2.0 | * Display: “Welcome to Stamford Academic Help Chatbot!”. * Display a menu of services such as:  1. Ask a general question (FAQs) 2. GPA Calculator 3. Course Finder 4. Academic Calendar 5. Budget Planner 6. Academic Guidance 7. Exit |  |

|  |  |  |
| --- | --- | --- |
| **Step (3.0)** | | **Objective: Get User Choice** |
| **Algorithm** | **Procedure** | **Output (Screen Shot)** |
| 3.1 | * **Read user input**   **If 1 → Ask a Question**   * Ask: "Enter your question" * Match with common keywords * Show relevant answer |  |
| 3.2 | * **Read user input**   **If 2 → GPA Calulator**   * Ask for number of courses * For each course, ask for grade and credit * Calculate GPA and display |  |
| 3.3 | * **Read user input**   **If 3 → Course Finder**   * Ask for a keyword that describe what user interested and expected to learn. * Find and show the result based on input keyword. |  |

|  |  |  |
| --- | --- | --- |
| 3.4 | * **Read user input**   **If 4 → Budget Planner**   * Ask for income and expenses * Calculate and show remaining balance |  |
| 3.5 | * **Read user input**   **If 5 → Academic Calendar**   * Display important dates (semester start, holidays, exam week) |  |
| 3.6 | * **Read user input**   **If 6 → Exit**   * Display: “Goodbye!” * End program |  |

# Screenshots / Sample Outputs

Insert screenshots or sample output from your program execution here.

# Conclusion

Summarize your experience and what you learned through this project.

# References

List any resources or references you used during the development.