**Certificate of Completion**

This is to certify that [Ibeawuchi Cornelius Chukwuemeka],[Ukwuezi Chinedu Junia], [Justus Chimezuru Uwalaka] has successfully completed the NumberConversion project on [3-Nov-2023].

[group E-project]

**Table of Contents**

* [Problem Definition](https://chat.openai.com/c/977d4c8e-55bc-4ce7-b595-a6a426a9699b" \l "problem-definition)
* [Customer Requirement Specification](https://chat.openai.com/c/977d4c8e-55bc-4ce7-b595-a6a426a9699b" \l "customer-requirement-specification)
* [Project Plan](https://chat.openai.com/c/977d4c8e-55bc-4ce7-b595-a6a426a9699b" \l "project-plan)
* [E-R Diagrams](https://chat.openai.com/c/977d4c8e-55bc-4ce7-b595-a6a426a9699b" \l "e-r-diagrams)
* [Algorithms](https://chat.openai.com/c/977d4c8e-55bc-4ce7-b595-a6a426a9699b" \l "algorithms)
* [GUI Standards Document](https://chat.openai.com/c/977d4c8e-55bc-4ce7-b595-a6a426a9699b" \l "gui-standards-document)
* [Interface Design Document](https://chat.openai.com/c/977d4c8e-55bc-4ce7-b595-a6a426a9699b" \l "interface-design-document)
* [Task Sheet](https://chat.openai.com/c/977d4c8e-55bc-4ce7-b595-a6a426a9699b" \l "task-sheet)
* [Project Review and Monitoring Report](https://chat.openai.com/c/977d4c8e-55bc-4ce7-b595-a6a426a9699b" \l "project-review-and-monitoring-report)
* [Unit Testing Check List](https://chat.openai.com/c/977d4c8e-55bc-4ce7-b595-a6a426a9699b" \l "unit-testing-check-list)
* [Final Check List](https://chat.openai.com/c/977d4c8e-55bc-4ce7-b595-a6a426a9699b" \l "final-check-list)

**1. Problem Definition**

The NumberConversion project aims to provide a set of utility functions for converting numbers between different bases, including decimal, binary, and hexadecimal representations.

**2. Customer Requirement Specification**

The customer's requirements for the NumberConversion project are as follows:

* The application should allow users to convert decimal numbers to binary, binary to decimal, decimal to hexadecimal, and hexadecimal to decimal.
* Users should also be able to perform binary addition.
* The application should provide a user-friendly interface for input and output.

**3. Project Plan**

The project plan for NumberConversion is as follows:

* Start Date: [24-Oct-2023]
* End Date: [15-Nov-2023]

Milestones:

* Design and Implement Conversion Functions:
  + Responsible: [Ibeawuchi Cornelius Chukwuemeka],[Ukwuezi Chinedu Junia],[Justus Chimezuru Uwalaka].
  + Deadline: [15-Nov-2023]
* Create User Interface:
  + Responsible: [Group]
  + Deadline: [15-Nov-2023]
* Testing and Debugging:
  + Responsible: [Testing Team]
  + Deadline: [15-Nov-2023]

**4. E-R Diagrams**

project dosen’t involve database

**5. Algorithms**

The following algorithms were implemented in the project:

\* decimalToBinary(int num)

    \* Description: Converts a decimal number to its binary representation.

    \* Implementation: `return Integer.toBinaryString(num);`

\* binaryToDecimal(String binary)

    \* Description: Converts a binary number to its decimal representation.

    \* Implementation: `return Integer.parseInt(binary, 2);`

\* decimalToHexadecimal(int num)

    \* Description: Converts a decimal number to its hexadecimal representation.

    \* Implementation: `return Integer.toHexString(num);`

\* hexadecimalToDecimal(String hexadecimal)

    \* Description: Converts a hexadecimal number to its decimal representation.

    \* Implementation: `return Integer.parseInt(hexadecimal, 16);`

\* binaryToHexadecimal(String binary)

    \* Description: Converts a binary number to its hexadecimal representation.

    \* Implementation:

    ```java

    int decimal = Integer.parseInt(binary, 2);

    return Integer.toHexString(decimal);

    ```

\* hexadecimalToBinary(String hexadecimal)

    \* Description: Converts a hexadecimal number to its binary representation.

    \* Implementation:

    ```java

    int decimal = Integer.parseInt(hexadecimal, 16);

    return Integer.toBinaryString(decimal);

    ```

\* binaryAddition(String binary1, String binary2)

    \* Description: Performs binary addition on two binary numbers.

    \* Implementation:

    ```java

    int decimal1 = Integer.parseInt(binary1, 2);

    int decimal2 = Integer.parseInt(binary2, 2);

    int sum = decimal1 + decimal2;

    return Integer.toBinaryString(sum);

    ```

---

These algorithms form the core functionality of the NumberConversion project, enabling conversions between decimal, binary, and hexadecimal representations.

**6. GUI Standards Document**

The GUI Standards Document outlines the guidelines and standards for the graphical user interface of the NumberConversion application.

* Use a clean and intuitive layout.
* Provide clear labels and input fields for user input.
* Display results in an organized and readable format.

**7. Interface Design Document**

The Interface Design Document provides a detailed description of the user interface for the NumberConversion application.

* Input Fields:
  + Decimal Number
  + Binary Number
  + Hexadecimal Number
* Buttons:
  + Convert Decimal to Binary
  + Convert Binary to Decimal
  + Decimal to Hexadecimal
  + Hexadecimal to Decimal
  + Binary to hexadecimal
  + Hexadecimal to Binary
  + Binary Addition

**8. Task Sheet**

The following tasks were assigned to team members:

* Task 1: Implement a program which performs Binary, Decimal and Hexadecimal Conversion.
  + Responsible: [Ibeawuchi Cornelius Chukwuemeka]
  + Deadline: [Date]
* Task 2: Design UI layout.
  + Responsible: [Team Member]
  + Deadline: [15-Nov-2023]

**9. Project Review and Monitoring Report**

**Project Progress Summary:**

The NumberConversion project has made significant progress since its initiation. The team has successfully implemented the core conversion algorithms, designed and integrated a user-friendly interface, and conducted thorough testing to ensure the application's functionality.

**Accomplishments:**

* Implemented all required conversion algorithms (decimal to binary, binary to decimal, etc.).
* Designed and integrated a clean and intuitive user interface for input and output.
* Conducted extensive unit testing to validate the correctness of conversion functions.
* Met key milestones according to the project plan.

**Issues Encountered:**

* **Input Validation**: There were occasional issues with input validation, particularly when users entered invalid characters or symbols. This led to unexpected behavior in some cases.
* **UI Responsiveness**: On certain devices with smaller screens, the user interface experienced minor responsiveness issues. Some elements did not display optimally.
* **Error Handling**: The application needs to incorporate better error handling mechanisms, especially when users provide incorrect or out-of-range inputs.

**Solutions:**

* **Input Validation**: Implemented additional checks to ensure that user inputs are valid. Improved error messages and provided clear instructions for valid input formats.
* **UI Responsiveness**: Conducted UI testing on various screen sizes and made adjustments to ensure optimal display on different devices.
* **Error Handling**: Enhanced error handling by providing informative error messages and instructions on how to correct input errors.

**Future Recommendations:**

* Consider adding additional features such as base conversion for octal and base64 representations.
* Explore the possibility of internationalization for broader user accessibility.
* Continuously monitor user feedback for further improvements.

**10. Unit Testing Check List**

The following unit tests were performed:

* Test decimal To Binary:
  + Input: 10
  + Expected Outcome: "1010"

Result: The test passed sucessfully.

* Test binary To Decimal:
  + Input: "1010"
  + Expected Outcome: 10

Result: The test passed sucessfully.

* Test Decimal To Hexadecimal:
  + Input: 255
  + Expected Outcome: ‘FF’

Result: The test passed sucessfully.

* Test Hexadecimal To Decimal:
  + Input: 2F
  + Expected Outcome: ‘47’

Result: The test passed sucessfully.

* Test Binary To Hexadecimal:
  + Input: “11011010”
  + Expected Outcome: ‘DA’

Result: The test passed sucessfully.

* Test Hexadecimal To Binary:
  + Input: “4C”
  + Expected Outcome: ‘1001100’

Result: The test passed sucessfully

* Test Binary Addition:
  + Input: “1101” + “1011”
  + Expected Outcome: ‘11000’

Result: The test passed sucessfully.

**11. Final Check List**

The final checks and validations performed before project completion are as follows:

* Ensure all conversion functions are working correctly.
* Verify user interface for usability and correctness.
* Conduct thorough testing and debugging.