**War Card Game Implementation**

**Overview**

The design and implementation strategy for creating a digital War card game are laid out in this proposal. The project's goal is to develop a playable, interactive game that follows the guidelines and goals of the conventional War card game. The Java programming language will be used for the implementation, and the provided base code will be expanded and customised to meet the unique needs of the game.

**Project Scope**

The project team consists of the following members:

- [Jasmine Saini- code Tester- Debuging]

- [Harleen Kaur - coder]

- [Priyanka Priyanka - coder]

- [Arman Sharma- Code Tester and Modifier]

The technical scope of the project includes:

- Developing a GUI (graphical user interface) to simplify user interactions.

- Executing the rules of the game, such as deck construction, card comparisons, battles, and wars.

- enabling two players to engage in simultaneous competition.

- Monitoring and showing each player's score as the game progresses.

- Selecting a winner when one player collects every card or when the game reaches its designated end time.

The completion of the project will be defined by:

- A GUI that is both practical and aesthetically pleasing and enables fluid gameplay.

- Correct application of the game's mechanics and rules.

- The choice to launch a new game or close the current one.

- Players' scores and game progress are displayed during play.

**High-Level Requirements**

The new system must include the following features:

- Players must register before the game begins.

- The sharing of game results, including victories and defeats.

- The score of each player is shown in real-time.

- A user interface that is simple and easy to utilise for fluid gameplay.

**Implementation Plan**

You may access the project's Git repository at <https://github.com/kau15021/Deliverable-1.git>. At the end of each day, developers will be required to check their code, assuring consistent updates and teamwork. Code, text files, and UML diagrams will all have their directories in the project's file structure.

The project will follow the following code guidelines:

- Consistent indentation and naming practices.

- For clarity, meaningful variable and method names.

- Use of comments in documentation appropriately.

- Effective use of the concepts of object-oriented programming.

The following equipment will be used by the development team:

- Integrated Development Environment (Name of IDE)

- Unit testing: [Name of the testing framework]

- Git for version control

**Design Considerations**

Some Object-Oriented (OO) ideas are demonstrated in the existing base code. Here are a few illustrations involving the base code:

1. Encapsulation:

Child classes must implement the abstract method "toString()" provided by the "Card" class. This captures the card-to-string conversion behaviour, which varies based on the particular card game implementation.

2. Delegation:

- The 'Game' class delegated to its subclasses the duty of implementing the game's behaviour. For example, the 'play()' method will be created in a child class to provide the War gaming mechanics.

3. Flexibility/Maintainability:

- The 'GroupOfCards' class offers a versatile framework for controlling a collection of cards in a game. By subclassing, it is possible to adapt and reuse this class to the needs of other card games.

To ensure a well-structured and maintainable codebase, the team will work to strengthen these principles during project execution while also adhering to other crucial design concerns.

**Conclusion**

The goals, parameters, specifications, and strategy for developing a digital version of the War card game are described in this proposal. Following this suggestion will provide the development team with a clear road map for making an entertaining and useful game that captures the spirit of the original card game. To create a top-notch gaming experience, the project will build upon the base code that has been provided.