**Final Report**

The Ceramics Inventory Management System is a program developed in C++ to help ceramic artists and fans manage and monitor their creations as they move through different production phases like bisque firing and glazing before final firing in the kiln. I came up with the concept for this project based on my passion for ceramics and the difficulties I faced in keeping tabs on each piece’s progress along with its clay composition and distinct characteristics. Users can input information about each ceramic item such, as size and weight and categorize them as either functional or sculptural pieces based on their texture and type of clay used in creation using a user friendly command line interface to handle the inventory efficiently.

The project is divided into two categories. CeramicPiece and Inventory classes that handle different aspects of the ceramics collection management system. The CeramicPiece class defines the characteristics of each piece like its name, stage, category texture and type of clay. Additionally it includes features for presenting details and storing/retrieving data in CSV format. On one hand the Inventory class oversees a group of CeramicPiece objects using a vector structure and provides functions for adding new pieces, displaying all pieces, filtering by attributes, and storing/retrieving inventory data to/from files. The functions command line interface provides a menu that is easy to use, allowing users to explore different options and interact with the inventory effortlessly.

Important components of the projects programming include:

1. To store and retrieve inventory data in CSV format and maintain data continuity, between sessions the streams ifstream and ofstream are utilized.
2. Each class, in the project is organized into individual.cpp and .h files to promote encapsulation and modular programming principles.
3. Using data structures, like the vector container enables the storage of pieces and facilitates flexible inventory management.

One of the difficulties I faced was dealing with file input/output and making sure that data is stored consistently over time in the system’s memory bank. To ensure that ceramic piece data could be saved and restored accurately as required by the system, I opted for a CSV file format that facilitates both parsing and storage processes. Dealing with exceptions in file input/output operations, like absence of files, wrong data structures, empty files demanded thorough validation, and error management to guarantee seamless user experience functionality.

During this project experience I developed more effective ways to work with file streams. Handling CSV data demanded techniques for parsing and converting data types highlighting the importance of strategic data management. This project underscored the significance of managing errors and enhancing user experience in command line tools. I came to understand that even basic text instructions can cause confusion if not carefully crafted. Through refining the menu layout and incorporating error notifications I developed a deeper respect for user focused design even within a text driven environment.

In the future of us I aim to improve this project by including:

* Enhancing Functionality; Include sorting features for users to arrange items based on characteristics such, as weight, dimension and phase.
* Add a user graphical interface (GUI) to enhance the accessibility and visual appeal of the application. Particularly for users who may not be familiar, with command line interfaces.
* Include data categories like the date of creation and approximate firing temperature to enhance tracking, for ceramic items more thoroughly.
* Add glaze as one of the options and allow the user to add their own glazes to the list.

The Ceramics Inventory Management System has been a fulfilling endeavor that enabled me to merge my passion for ceramics with learning the C++ programming language. Developing the inventory system provided me with practical knowledge in object oriented design principles as well as file handling and modular programming techniques.