**Milestone Four**

**Enhancement 3 Narrative**

**Databases**

**Justin Byrd**

**justin.byrd@snhu.edu**

**CS-499-H7089 Computer Science Capstone**

**Professor Brooke Goggin**

**12/3/2023**

# Briefly describe the artifact. What is it? When was it created?

The artifact central to my enhancements is the "RGB LED Cube Control Software," a project initiated during my Electrical Engineering Technology Associate degree program's capstone in 2015. Initially, I constructed a 4x4x4 LED Cube with RGB LEDs, shown below, integrating it with an FPGA development board and crafting custom firmware to manipulate the 52 control lines governing the 64 LEDs.

A green box with wires

Description automatically generated

In a recent pursuit of rekindling my enthusiasm for the project, I designed additional circuitry to interface my RGB LED Cube Control Software with an Arduino Uno, facilitating control over the RGB LEDs. Prior to the start of this course, I had developed a simple functional user interface (UI), shown below, establishing communication with the Arduino Uno through a COM Port. This initial UI empowered me to illuminate one color of one LED based on user-defined color, X, Y, and Z values.

A screenshot of a computer screen

Description automatically generated

Having accomplished the hardware integration and circuitry enhancements, the focus of my Computer Science Capstone project pivots decisively to the evolution of the RGB LED Cube Control Software. With the foundational hardware components in place and the initial user interface demonstrating functional control over individual LEDs, my aim is to delve into advanced features and optimizations within the software realm. This shift marks the latest chapter in the ongoing development of a project that seamlessly marries my technical skills in both hardware and software domains.

# Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in software development? How was the artifact improved?

The RGB LED Cube Control Software finds a prominent place in my ePortfolio, justifiably spotlighting the transformative enhancements realized during the third development phase. This phase delves into the intricacies of back-end database integration, highlighting my prowess in architecting comprehensive full-stack applications.

## Database Tab Functionality:

A screenshot of a computer

Description automatically generated

The augmentation of the Database Tab functionality stands as a testament to my expertise in graphical user interface design. The introduction of a ListView Widget, managing and implementing dynamic user interfaces effectively, showcases a keen eye for user interaction and functionality.

## New Class to Support Database Interaction - sqliteCRUDModule Class:

A screenshot of a computer program

Description automatically generated

A significant architectural leap is marked by the introduction of the sqliteCRUDModule class, a dedicated module for database interaction. This class not only encapsulates the complexities of database connections but also exemplifies my commitment to modular and organized code. The constructor and destructor operations ensure the efficient handling of database resources, mitigating potential memory leaks.

## Member Function Additions in Existing Classes:

### RGBLEDCCS Class Member Function Additions:

A black background with white text

Description automatically generated

### cubeControl Class Member Function Addition:

A black background with white text

Description automatically generated

Incorporating new member functions into existing classes, such as RGBLEDCCS and cubeControl, reflects the seamless integration of software enhancements into a pre-existing codebase. These additions act as private slots and member functions, ensuring a harmonious synergy between UI functionality and the underlying database operations.

## Database Creation:

A computer screen shot of a number

Description automatically generated

The creation of the 'RGB\_LED\_CCS\_DB.db' database, housing a table named 'cubePatterns,' is a deliberate step towards structured data storage. The table, with columns 'patternName' and 'patternString,' sets the foundation for efficient application functionality. This meticulous structuring underscores the significance of organized data management in the context of software design.

## Video:

A visual walkthrough encapsulated in the video linked below provides a dynamic exploration of these updates. The video not only discusses the intricacies of the software enhancements but also showcases the live functionality of the RGB LED Cube Control Software.

Link to video hosted on YouTube - [Enhancement 3](https://www.youtube.com/watch?v=LhghKqOzzrc)

# Did you meet the course objectives you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?

In pursuit of the course objectives, the enhancements made to the RGB LED Cube Control Software align with and contribute to the broader goals of the course outcomes.

## Outcome One:

The first outcome I met with this enhancement was, “Demonstrate and ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals”.

### UI Design and Implementation:

The augmentation of the Database Tab functionality serves as a tangible manifestation of my expertise in graphical user interface design. The introduction of a ListView Widget not only enhances user interaction but also showcases a dynamic and effective implementation of graphical interfaces.

### Encapsulation and Modularity:

The creation of the sqliteCRUDModule Class epitomizes my commitment to encapsulation and modularity in software development. Employing dummy functions initially allowed for incremental testing and updates, enabling a systematic integration process. This approach not only streamlined development but also showcased a methodical approach to modularizing functionality.

### Using Industry Tools to Facilitate Efficient Work:

Acknowledging the challenges faced with deploying MongoCXX drivers, which I will talk about in more detail below, the decision to pivot towards SQLite demonstrates a pragmatic approach to problem-solving. Leveraging the readily available SQL library in QT reflects an understanding of industry tools, facilitating efficient development and avoiding unnecessary complexity.

## Outcome Two:

The second outcome I met with this enhancement was, “Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution, while managing the trade-offs involved in design choices”.

### Managing Trade-offs:

The decision to switch from MongoDB to SQLite during this enhancement highlights the practical aspects of managing trade-offs. Balancing the robustness of databases with project deadlines necessitated a strategic shift, emphasizing the importance of adaptability and decision-making in a professional context. This scenario mirrors real-world situations where project scope adjustments are made to meet schedule requirements.

### Pattern Generation Algorithm:

The implementation of a pattern generation algorithm in the RGB LED Cube Control Software exemplifies a commitment to algorithmic principles, emphasizing both time complexity and optimization. Employing nested loops to iterate through all 64 LEDs in the static 4x4x4 RGB LED Cube, the algorithm achieves a time complexity of O(n^3), ensuring efficiency for the fixed cube size. To optimize the process, getter functions of the LED objects in the cubeControl class are utilized, offering controlled and encapsulated access to color values. Despite not directly accessing the member variable, this design choice ensures data integrity while facilitating a streamlined pattern generation. The algorithm's careful balance of time complexity and optimization adheres to best practices, reflecting a thoughtful approach to performance considerations within the RGB LED Cube Control Software project.

## Outcome Three:

The third outcome I had planned to meet in this enhancement but had to pivot on was, “Develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data resources”.

### Explanation Outcome was not Met:

While the initial plan involved MongoDB and its user access functionality, a shift to SQLite occurred due to practical considerations and deployment challenges. Although this pivot led to the exclusion of user access controls in this enhancement, it's crucial to note that the broader security objective was already met in Enhancement 1 with the implementation of a secure login page. The decision to prioritize project completion and maintain overall security aligns with a pragmatic approach to software development.

# Reflect on the process of enhancing and/or modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?

## What I Have Learned:

### CMake:

The journey of integrating MongoDB drivers introduced me to the powerful world of CMake. While the ultimate goal of successfully building the MongoCXX drivers remained elusive, delving into CMake provided valuable insights into its functionality. This encounter revealed CMake's role as a robust tool for building diverse solutions based on various variables, such as compilers and other available libraries. Despite the specific challenge faced, the exposure to CMake proved to be an enriching learning experience.

### SQLite Database:

Navigating the terrain of SQLite, a lightweight SQL-type database, presented an opportunity to refresh and deepen my database knowledge. Having taken a class on MySQL databases in the past, reacquainting myself with database interactions through SQLite required significant research. This step served as an informative immersion into SQLite's intricacies, adding a new layer to my understanding of database management within the project.

### Working with QT Models and ListView Widget:

The decision to incorporate a ListView Widget into the project marked a departure from familiar QT widgets. To populate this widget with pattern names from the database, delving into QT models and understanding their role in dynamically sizing database tables for ListView widgets became a crucial aspect of the learning process. This exploration expanded my toolkit within the QT framework and broadened my understanding of more intricate UI elements.

## Challenges Faced:

### MongoDB Integration Issues:

The initial pursuit of integrating MongoDB with C++ software posed significant challenges. While I achieved success in running MongoDB on my system and interacting with it through the command line using Mongo Shell, building the crucial MongoCXX drivers proved to be a formidable hurdle. The process of acquiring and building the drivers led to multiple roadblocks, encountering build errors and complications specific to my compiler. Despite extensive research and attempts, a resolution remained elusive within the project timeline.

### Transition to SQLite:

In response to the MongoDB integration challenges, a pivotal decision was made to pivot towards SQLite as an alternative database solution. This shift brought its own set of challenges, particularly in the realm of interacting with SQL databases. Unfamiliarity with SQLite necessitated in-depth research and a learning curve, but the decision proved fortuitous as it allowed for the successful completion of the enhancement within the project constraints.

## Conclusion:

The enhancement process was marked by dynamic learning experiences and strategic decision-making. From grappling with CMake intricacies and navigating SQLite's nuances to incorporating advanced QT widgets, each challenge presented an opportunity for growth. The ability to pivot from MongoDB to SQLite underscored adaptability and resilience, showcasing the real-world decision-making inherent in software development projects. Overall, the journey encapsulated not just technical advancements but also the development of problem-solving skills and a deeper understanding of database management within the project context.