**Semester Project**

Jonathan Torres

School of Business, Liberty University

CSIS312: Advanced Object-Oriented Programming

Professor Festus Oderanti

April 30, 2023

**Introduction**

JavaFX-based alarm clock applications have gained popularity in recent years due to their simplicity and ease of use. In particular, the basic clock project offers a well-built and functional application that serves as an excellent starting point for further development. My experience with JavaFX is primarily limited to in-class assignments, with the Contact app being the most challenging and educational project I have worked on thus far. I wanted to apply the knowledge gained from these assignments to a new project, ultimately choosing to contribute to the basic clock project.

The main goals of my contribution to this project were to clean up the existing code, apply best practices, and enhance a user interface with new features. These changes included adding a graphical interface for the clock, incorporating an alarm and timer functionality, and improving the overall user experience. More minor but still important features on my task list included polling time from an NTP server, converting between military (24 hour) and standard 12-hour time, allowing the user to choose if AM or PM was visible in the 12-hour time format, and providing support for time zones and custom alarm sounds. Most influentially, I wanted the clock to at times in which the time matched verses of inspiration, to display that verse on the graphical interface.

The biggest challenge I faced was understanding the logic and reasoning behind the original codebase in order to extend it in a way that made it more extensible. In the process, I aimed to not only improve the basic clock application but also to further my skills in JavaFX and software development in general. Throughout this paper, I will discuss my experiences working on this project, the steps I took to contribute, and the overall impact of my work on the application and its potential users.

**Biblical Integration**

Time is a significant aspect of our lives and profoundly impacts our actions, decisions, and spiritual growth. In Ecclesiastes 3:1 it says, "There is a time for everything, and a season for every activity under the heavens." This verse highlights the importance of managing and understanding time daily. The basic clock application, with its new features and added functionality, can serve as a tool for reminders for individuals to reflect on their relationship with time and, subsequently, with God.

The application's ability to set alarms, timers, and time zone adjustments might encourage users to be more intentional and mindful with their time, using it to align with their spiritual beliefs and values. By prompting users to contemplate the significance of time in their lives (setting the alarm/timer), the clock application can be a symbol as a proxy of one's spirituality, ultimately contributing to the expansion of God's kingdom and cultivating more meaningful, faith-centered lives.

Additionally, the project can be utilized to serve various organizations and institutions that prioritize faith and spirituality. Churches, for example, can use the application's alarm and timer functionalities to facilitate prayer sessions, Bible studies, and other spiritual gatherings. By offering an efficient tool that enhances time management and organization, the clock application can play a role in promoting spiritual growth and deepening the connection between individuals and their faith. More to an individual, it was hoped that this application eventually would be able to be used and enjoyed for the scripture that would periodically appear on the interface.

The clock application can serve as a reminder of the temporary nature of life, urging users to seek purpose and meaning in their actions and decisions. By encouraging individuals to reflect on the significance of time and their relationship with God, the application supports the cultivation of more intentional, spiritually rich lives. This focus on enhancing the clock application's features aligns with our exploration of meaningful contributions to the project, such as the implementation of a specific feature or addressing an issue that impacts the overall user experience.

Graphical Interface Implementation and Pull-Request

The primary feature I chose to focus on for this project was the graphical interface. Enhancing the visual aspects of the application would significantly improve user experience, making it more intuitive and enjoyable for users to interact with. This section will detail my process for implementing the graphical interface, from the initial analysis of the existing code to the ongoing progress on the pull-request. This discussion spans over two pages, providing a comprehensive account of my experience contributing to this open-source project.

Initial Analysis

Before implementing the graphical interface, I carefully analyzed the existing code to understand the overall structure and design of the application. I identified the areas that would require modification or improvement and outlined a plan for implementing the necessary changes. This process involved researching JavaFX best practices and familiarizing myself with the relevant libraries and tools.

Designing the Interface

With a clear understanding of the existing code and the desired outcome, I began designing the new graphical interface. I created wireframes and mock-ups to visualize the layout, and iteratively refined the design based on feedback from my peers and other contributors to the project. The goal was to create an interface that was both visually appealing and easy to navigate, ensuring that users could effortlessly manage their alarms and timers.

Implementing the Interface

After finalizing the design, I proceeded to implement the graphical interface using JavaFX. I carefully integrated the new interface components with the existing application logic, making sure that the new interface worked seamlessly with the underlying functionality of the clock and alarm system. Throughout the implementation process, I remained mindful of best practices, such as using proper coding conventions, adhering to the Model-View-Controller (MVC) pattern, and organizing the code in a modular and maintainable manner.

Testing and Refinement

Once the implementation was complete, I thoroughly tested the new graphical interface to ensure that it functioned correctly and did not introduce any bugs or issues into the application. This involved testing various use cases, such as setting and deleting alarms, interacting with the timer functionality, and adjusting timezone settings. I also sought feedback from other contributors and users, refining the interface based on their suggestions and observations.

Documentation and Pull-Request

With the graphical interface successfully implemented and tested, I documented my changes, including the issue I addressed, the modifications I made to the code, and the testing process I followed. I also provided screenshots and other visual aids to demonstrate the improvements made to the application.

Finally, I submitted a pull-request to the original project repository, requesting that my changes be reviewed and considered for merging with the main codebase. In my pull-request, I included a detailed description of the graphical interface I implemented, the reasons behind my proposed changes, and any additional information that would help the project maintainers understand and evaluate my contribution.

Ongoing Work

It is important to note that the project is not yet completed. While significant progress has been made on the graphical interface, there may be additional refinements and enhancements required before the feature is fully integrated into the application. I am committed to collaborating with the project maintainers and other contributors to ensure that the graphical interface is polished and meets the needs of the user community.

# Bibliography

Life.Church. (2022). *YouVersion*. Retrieved 08 25, 2022, from https://my.bible.com/bible/111/PRO.18.NIV