Apr 20th, 1:00 PM - 3:00 PM

Cryptic Coin Cloud

John Falcone

Cryptic Coin Cloud

By: John Falcone
Dr. Cenk Erdil & Dr. Frances Grodzinsky
Computer Science

Introduction

The world is turning digital in ways nobody imagined particularly with cryptocurrency. Bitcoin, a form of making payments with digital currency, proved that anything physical could get transformed into a digital format. Although, transitioning from physical currency to digital currency isn't without problems. Cryptocurrency gets obtained in two ways, the first way being a computational process called mining. In mining, users let their device perform necessary mathematical questions for generating new cryptocurrency in the network and securing the network through the device. Miners get compensated for lending their device's hardware capabilities by getting distributed cryptocurrency. The other method for obtaining cryptocurrency is more direct and involves using a digital currency exchange platform like Coinbase. These digital currency exchange platforms let users convert physical money on their credit card or bank account into cryptocurrency. In both cases, users get awarded with cryptocurrency signed with a user's private key for validating the transaction. Although, if this private key gets compromised through a damaged hardware or hacker stealing the file then the user will lose access to their crypto-currency. Cryptic Coin Cloud, a record management system application, would help remediate this problem by offering another location for backing up private keys and other relevant cryptocurrency data.

Users would have access to Cryptic Coin Cloud on a desktop, mobile device, or a web application. When the user first launches the application, they would be required to create an account using their e-mail address and password. Once a user logs in they can add records into six different tables, cryptocurrency information, credit card information, banking account information, password information, software license information, or contact information. Users also can view the records they entered into a table using select data from when they created the record. When users create a record, they will give it a unique record name and code for their entry into the table. The record name acts as a primary key used when establishing new records, picking a record to update, or removing the entire record from the table. The record code is a security measure used for verifying a user has permission for viewing their record and preventing unauthorized access.

The inspiration for Cryptic Coin Cloud is offering users an application where they could store records about cryptocurrency and related fields such as payment information. The design goals for this application is creating an easily accessible application all users could understand and implement. Cryptic Coin Cloud is beneficial for demonstrating how embedded databases interact with a front-end application and establishing connections to these databases for creating, modifying, viewing, or removing records. Cryptic Coin Cloud will get implemented using Java in the Netbeans Integrated Development Environment with embedded Apache Derby databases. Potential resources for future cloud implementation are Chameleon Cloud, Oracle Cloud, or Google Cloud. Future ideas for mobile implementation would involve creating an Android application. After reviewing different options for private clouds, CCC will use an OpenStack based cloud, Chameleon Cloud. OpenStack is will get used due to being both cost-effective and flexible for development on desktop, mobile devices, and a web application.

Research and Analysis

The first step in designing the application involved selecting a programming language for development. After researching various programming languages, Java got deemed the best candidate for developing the Cryptic Coin Cloud (CCC) application. Java is the ideal programming language for CCC due to its flexibility and efficiency as demonstrated by the laboratory information management system for microarrays (LIMaS) developed by Mammalian Genetics Unit. LIMaS uses a Java-framework and has an "efficient and fully customizable interface giving it the ability to adapt to any working practice, e.g., handling many resources used to form many products (chaining of resources)" (Webb et al.). CCC stores records in multiple tables in a database so it is important having a program that could manage multiple tables at once without performance issues. Another reason Java got chosen for CCC's main language is Java's functionality on multiple devices. Java is a popular language used in many devices with Java getting "estimated to be running on over 3 billion devices worldwide. No other language runs on as many devices" (Vornick). A goal for CCC involves creating a desktop application, mobile application, and web application meaning that it must be accessible on multiple devices. Java is a great language for the application since it would assist with putting CCC on multiple devices. An alternative considered for CCC is programming it in PHP since that would be helpful when deploying the application on the cloud or another web application. Although, PHP didn't get selected for a few reasons consisting of vague error messages, susceptibility to SQL injections, and issues with the library APIs. These issues have workarounds; however, they would have added undesired delay to development time for the application.

After deciding a language, selecting a Java-based tool for the database became the next logical step. Apache Derby is an open-source relational database initially released in 1997 and used for creating embedded databases. Although the software is older, it still gets patched and got used for over twenty years making it a reliable choice for the application. CCC would benefit from using Apache Derby with it's databases "due to its limited footprint and ease of use" (Tsur). A limited footprint would be important for ensuring that CCC has good performance rate and remains a reliable back-up resources for its users. Another benefit of developing databases with Apache Derby is that it has flexibility allowing SQL executions getting incorporated which is beneficial for storing and retrieving records. Derby also gets used along with Java for creating "online databases such as MalaCards, Biomodels and NCBI's databases" (Tsur). Since a goal for CCC is creating web application, having a database manager capable of hosting on local and web applications is perfect. Initially an alternative considered for the database manager was using NoSQL; however, since NoSQL has difficulties with keeping table entries consistent which would get problematic when a person must rely on them as a back-up record. In addition, NoSQL contains many security vulnerabilities which are problematic when storing confidential information.

Netbeans proved a beneficial integrated development environment (IDE) since it contains options for developing applications using Java and Derby Apache. Netbeans got selected with CCC's accessibility in mind since the GUI editor in Netbeans got praised because "Java applications not only run correctly but also look right on a variety of devices with differing displays" (Coffee). When developing an application for users it is important the menus appear simple and organized which is perfect for CCC's simplistic menu design. Netbeans is also beneficial for developing a research project application due to it's ease of learning with "greater convenience and productivity in designing and building GUI applications" (Coffee). With a limited development time for working on CCC picking Netbeans as an IDE proved beneficial.

Other design choices considered for future implementation were deploying an application on the cloud and a mobile application. When picking a resource for cloud development, its important that the application had private cloud support. A private cloud is the best option for CCC since it offers more security than public clouds. Security in private clouds is stronger than public clouds due to public clouds allowing more access than a private cloud. Although mobile development is currently listed as a future idea, the ideal platform for mobile development would be Android since it offers better support for Java applications opposed to iOS.

References

S., Jothi, and Therese Avila. "Cryptography and Efficient Data Transmission for Cluster-Based Wireless Sensor Networks." International Journal of Multidisciplinary Approach & Studies, vol. 3, no. 5, 2016, pp. 34–45. Accessed 6 Sept. 2017.

K.A. Balygin, A.N. Klimov, A.V. Koirol'lov, S.P. Kulik, and S.N. Molotkov. "On Protection against a Bright-Pulse Attack in the Two-Pass Quantum Cryptography System." Pis'ma v Zhurnal Eksperimental'noi I Teoretichsekoi Fizki, vol. 102, no. 12, 2016, pp.883-890. Accessed 12 Sept. 2017

Vicentiu Radulescu. "Application Of Differential Cryptography To A GN Authentication Hiearchy Scheme." Electronic Journal of Differential Equations, vol. 2017, no. 20, 2017. pp 1-8. Accessed 18 Sept. 2017.

Zobel, Justin. Writing for Computer Science. 3rd ed., London, Springer, 2015.

Schneier, Bruce. Applied Cryptography Protocols, Algorithms, and Source Code in C. 2nd ed., New York, Katherine Schowalter, 1996.

Stanoyevitch, Alexander. Introduction to cryptography with mathematical foundations and computer implementations. Boca Raton, Chapman & Hall, 2011.

Geyer, Ryan. "Build a Private Cloud in Your Garage." Build a Private Cloud in Your Garage, 8 Mar. 2012, www.rightscale.com/blog/cloud-management-best-practices/build-private-cloud-your-garage. Accessed 26 Sept. 2017.

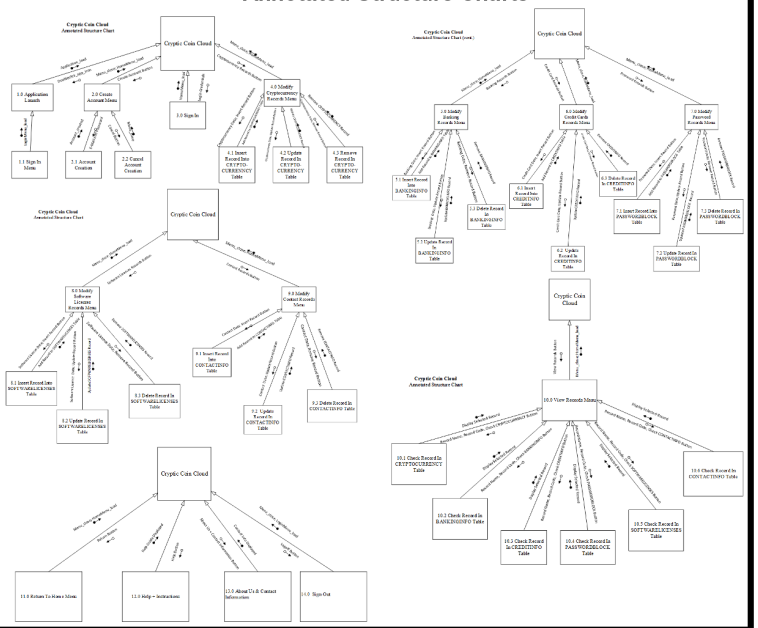
Webb, Sarah, Attwood, Anthony, Brooks, Tony, Freeman, Tom, Gardner, Phil, Pritchard, Clare, Williams, Debbie, Underhill, Peter, Strivens, Mark, Greenfield, Mark, Pilicheva, Ekaterina "LIMaS: the JAVA-based application and database for microarray experiment tracking." PLOS Computational Biology. 14th ed. 12 Mar. 2018. Accessed 14 Dec. 2017.

Vonick, Jim. "10 Reasons Why You Should Consider Learning Java." 10 Reasons Why You Should Consider Learning Java | Oracle University Blog, blogs.oracle.com/oracleuniversity/10-reasons-why-you-should-consider-learning-java. 29 Feb. 2016. Accessed 16 Dec. 2017.

Tsur, Elizabeth. "Rapid development of entity-based data models for bioinformatics with persistence object-oriented design and structured interfaces." BioData Mining, Vol 2017. Accessed 16 Dec. 2018.

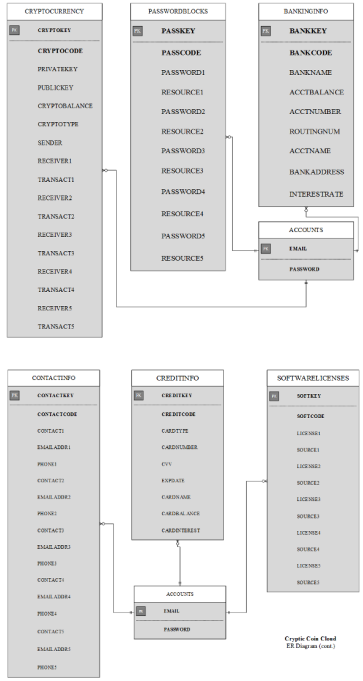
Coffee, Peter. "NetBeans 5.0 makes 'free' look good." eWeek, Vol. 23 Issue 7, p52-53. 13 Feb. 2006. Accessed 17 Dec. 2017

Annotated Structure Charts



Interface Screenshots

ER Diagrams



Conclusion

Developing the Cryptic Coin Cloud application provided a valuable lesson in software production. There were many challenges at the beginning of development consisting of not having a set project in mind to extra labor due to working alone. Although, these challenges were present and there were moments when it felt daunting, I never made sure to give up hope. I combatted the first challenge of not having a set idea by picking a topic like Cryptocurrency which had a broad scope and I could narrow it down from there. My approach for picking a general topic did cause more challenges later when I had to narrow down the scope from a cryptocurrency wallet to a record management system; however, it worked out. The CCC experience wasn't an entirely negative experience though as it had positive moments.

One benefit for working on this application is that it exposed me to the challenges of building software which is an experience I had before, but not to the degree this class offered. In addition, building this application helped re-enforce database design. Although, I did have experience with working on databases from a class I took here, I never felt my knowledge of databases were the strongest. There are many database concepts that I still should learn more about; however, I feel more confident about using database design with Apache Derby after this experience. I'm also glad this project helped teach me Java since it's a popular language used in many places which is beneficial for future development projects. In addition, the project taught me challenges with documentation because while the development felt more challenging than documentation, doing all the documentation for Cryptic Coin Cloud taught me how time-consuming and precise it could get.

Overall, there were many challenges when developing Cryptic Coin Cloud and working alone proved troublesome when reaching the final project; however, it helped make me realize the challenges in software development. With these challenges in mind, I strive towards improving the Cryptic Coin Cloud application and providing better products in the future. The project did help with establishing syntax and concepts for Java, Apacche Derby, and learn how to use Netbeans, a different integrated development environment than Visual Studio. There are many obstacles for reaching better quality programming, but Cryptic Coin Cloud is a good introduction to the realities behind software development.