Neighborhood Informant - Final Summary

Group 2: Aiwan Hazari, Deven Patel, Dhrumil Patel, and Jay Patel

---

Project Summary

Neighborhood Informant is a desktop application that provides ease to Chicagoland residents by providing a single application of various Chicagoland data. This application will gather real and accurate data straight from the City of Chicago data portal website. There will be an easy-to-use user interface that will allow a user to pick a location and neighborhood and receive relevant feedback about it, such as crime, landlords, etc. This application can be used by anyone from anywhere, and its simple GUI makes it easy to navigate without much support. Neighborhood Informant will be a one-stop for a multitude of accurate Chicagoland information.

Tools

For development and collaboration, we mainly used Asana, Git, WhatsApp, Firebase, and Eclipse. Asana is an easy to use website application that allows group members to work together and see the project status. Here we assigned tasks and planned our scripts. Git was used to store our code and update it for all team members to use. WhatsApp was used to communicate with everyone on the team about the project, assignments, and meetings.

Firebase is a database application and we used it to store all of our data, as well as update data onto our application in real-time. Eclipse was used for our main codebase production. Eclipse had easy to use tools, such as Window Builder, to implement the user interface using the Java Swing library.

Maven is used to automatically build some of the functionality of the project. The tool enabled us to not use external .jar files for Firebase. Instead, we used Maven build for it. It worked flawlessly. Everything started communicating with each other.

Requirements

This project needs a (Windows, Mac, or Linux) machine with at least 2GB of memory. It also needs the latest version of Java installed. The machine's Wi-fi also needs to be on and connected. Further, a JXBrowser license is required for use, which can be download via the JXBrowser website itself. When including all the .jar files from JXBrowser, the user needs to include the license.jar file to replace the old one. After doing this, the JXBrowser should work without any problems.

Programming Practices

Agile development, in particular Extreme Programming, was mainly used for development. Test-Driven Development and Pair Programming were also used concurrently.

Testing

For testing, JUnit testing was used early in the development phase, in order to prevent refactoring of the code. It was quite efficient at the end because not a lot of changes were needed in the final stages. On top of that, continuous tests were performed half the time of every sprint. Final testing was also done in the fifth sprint, which was allocated as the testing sprint.

Preview of Project/Graphical User Interface:

More details on GUI and its functionality is in the final report.