**WEEKLY LOG – SAFEWAYZ MOBILE APPLICATION**

|  |  |  |
| --- | --- | --- |
| Week | Progress | Next steps |
| 1 | * Research possible ideas and find datasets | * Select an idea and find solid data to support it |
| 2 | * Selected a project name and idea * Found data of San Francisco crime streets * Assign team roles and split up the proposal document and presentation | * Create the PowerPoint presentation and complete proposal document |
| 3 | * Proposal document completed along with powerpoint presentation | * Experiment with google maps API * Find a data set with all streets in San Francisco * Merge crime and non-crime data set into a single JSON file |
| 4 | * Experimented with google maps API * Found a dataset with all streets in San Francisco * Merged crime and non-crime data set into a single JSON file | * Find a graph traversal algorithm to see if JSON graph can be traversed (DFS OR BFS). * Test traversal algorithms * Develop an equation for weightings and JSON for the weighting types and crime frequency |
| 5 | * Developed DFS & BFS traversal. * Cross-checked the paths to see if valid on Google Maps * Developed a mathematical eqn for the weighting based on the weighting amounts corresponding to crime type and date in JSON. | * Plan/Develop possible Dykstra’s algorithm for optimal route from crimes. * Complete requirements specification document |
| 6 | * Finished the Software Requirements specifications along with updated proposal document * Planned how to setup Dykstra’s algorithm | * Try to create a module to implement Dijkstra’s algorithm. * Research details on implementing DFS to find all paths instead of one unique path. |
| 7 | * Completed BFS & DFS to save all possible paths to a destination. * Partial working Dijkstra’s algorithm with HashMaps instead of LinkedLists. * Revised and completed Software Requirements with feedback from Erik & Aida. | * Complete Dijkstra’s shortest path module * Complete unit testing and integration for currently completed modules |
| 8 | * Completed JUnit tests for majority of the modules. Dijkstra’s remains to be tested for. * Started some integration tests to see how the modules behave with each other | * Next steps, finish Dijkstra ASAP * Begin development of the mobile app by adapting the eclipse code to work for Android Studio |
| 9 | * Successfully implemented Dijkstra’s least crime algorithm. * Unit tests have begun on this. * Completed integration tests * Migrated eclipse java code for use in Android studio. * Development of base functionality such as DFS, crime search feature, and dikstras complete * GMaps waypoints API successfully tested and implemented to plot the given paths generated by DFS. * Significant improvements done on UI elements of the application | * Plotting of Dijkstra’s path * Clean up code comments and folder structure * Finish final touches on Requirements Specification report using feedback from Erik & Hollie * Begin working on Design Specification report * Create a controller/main activity for Eclipse Terminal implementation |
| 10 | * Cleaned up the folder structure of all the files on the repo * Code cleanup and detailed commenting added * Eclipse Terminal version of the application was successfully completed. * Android application is also mostly complete, but has some final steps left to plot Dijkstra’s correctly on the map(currently only prints the path) * Requirements specs document completed and submitted | * Complete the Design specs document * Add any final touches to android app * Update README.md file for instructions on android and eclipse terminal versions of the program |
| 11 | * Design Specifications document complete * Android application and terminal/console version are both complete * Video demonstrations and QuickStart guides documented. * README.md files updated |  |