Task Documentation

Reliability Analysis and Visualization for WARP

Objective: Analyze and visualize the end-to-end reliability of message transmissions within the WARP system

Maintainer: Nancy Nahra

Developers: Maria Gauna, Tommy Loi, William Lucas

Project TimeLine and Sprint Breakdown

- **Sprint 1: Due November 12**
  - Deliverables: High level plans, README.md Updates, UML Sequence diagram, preliminary design and project plans.
- **Sprint 2: Due November 22**
  - Deliverables**:** Updated UML diagrams, initial ReliabilityVizualization class, ReliabilityAnalysis JUnit tests and JavaDoc comments, README.md.
- **Sprint 3: Due December 13**
  - Deliverables: Completed ReliabilityAnalysis and ReliabilityAnalysis classes, Junit test, final UML diagrams updated, README.md

Key Components of Each Sprint

- **Sprint 1: Due November 12**
  - Deliverables:
    - README.md: Documenting task assignments and project status
    - UML Diagrams: Show program flow of 'ra' option in WARP
    - Design Documents: Design first preliminary project plans, artifacts that may be needed, UML diagrams, tasks, and project timeline.
- **Sprint 2: Due November 22**
  - Deliverables**:**
    - README.md: Updated with progress and team task assignments
    - Update UML Diagrams: Reflecting class and sequence with new methods.
    - ReliabilityVisualizaiton Class: Correct output/flow, higher-level helper methods, stepwise refinement to keep correct program flow
    - ReliabilityAnalysis Class: Initialize Class, create JUnit tests
    - Formatting: JavaDoc comments for new methods, follow Google style guide
- **Sprint 3: Due December 13**
  - Deliverables:
    - README.md: Final Documentation of task completion and project status.
    - Final UML Diagrams: Completed class and sequence diagrams.
    - ReliabilityVisualizaiton Class: Fully functional with JUnit Test and JavaDoc comments

- ReliabilityAnalysis Class: Fully functional with JUnit Test and JavaDoc comments