Revision History

Date	Version	Change	Author
02/14/2013	1.0	Initial Draft	Thomas Nelson
02/26/2013	1.1	Revised Draft	Thomas Nelson

Description

The MAPPING class is used to handle most of the navigation functions and is an extension of thread so that it can run in the background. Whenever the MAPPING class is called it will calculate a route between the user and a desired destination. The MAPPING class will be using the node list in the database and the ASTAR class to calculate an effective route. Finally this class will return the route in the form of an array of positions.

Class Variables

- **private Position userLocation;**
    - holds the position of the user
- **private Position destinationLocation;**
    - holds the position of the destination
- **private Position[ ] navigationRoute;**
    - holds the route in the form of an array of positions
- **private Handler parentHandler;**
    - Connection to the parent activity handler to send messages to
- **private Handler mappingHandler;**
    - Handles and responds to any messages from the parent activity
- **private Position previousLocation**;
    - Holds the users previous location if available

Constructors

- **public Mapping (Handler handler);**
    - creates a new thread
    - creates link to the parent thread's handler

Methods

- **private void calcUserLocation ( );**
  - call the LOCATE class to calculate the user location
- **private void calcDestinationLocation ( );**
  - Use the database to get all information about the destination node
- **public Position[ ] calcRoute (Position destination);**
  - call the ASTAR class to calculate the route with the given locations
- **private boolean checkWifi ( );**
  - checks the Wi-Fi status and returns it as a boolean value
- **public void run ( );**
  - overrides the thread run method
  - loops every 5 seconds to update user location
  - if user location changes then send message to activity handler
  - notifies parent activity if Wi-Fi signal is lost
- **public Handler getHandler ( );**
  - called by the parent activity to return the thread event handler
- **private void handleMessage (Message msg);**
  - handles any messages from the parent activity

Debugging Methods

- **void printUserLocation ( );**
  - prints all the user location information if available
- **void printDestinationLocation ( );**
  - prints all the destination information if available
- **void printRoute ( );**
  - prints the current route if available
- **void printWifiStatus ( );**
  - prints the status of the Wi-Fi

Class Responsibility Collaboration Card

Responsibilities	Collaborators
Holds user location	LOCATE
Holds destination location	POSITION
Gets location info from the database	ASTAR
Checks Wi-Fi status	Map Activity
Extends the thread class to run in the background	
Calls all other navigation classes	

Class Design Sign-off

Name	Approval Status
Matthew Menonkariyll	
James Grisdale	
Taras Mychaskiw	