CAMPS
Andrew, Charley, Dale, Jian, Justin, Ryan
“Design for speed!”
Customer Requests

• Design for speed

• Store the graph in memory to decrease response time

• Find which search algorithm is faster and use it (Dijkstra or A*)
Zero Feature Release

- Prototype Database up and running with small data set for testing
  - Data will consist of a small test set of nodes, buildings, and edges
  - Data will not include path type or transportation method
- Map image segments rendered and spliced up for retrieval by UI
- Servlet is running on Tomcat platform:
  - Can accept incoming JSON requests and decrypt into internal data structure for analysis
- Partially-developed UI:
  - Search Box allows building a list of destinations
  - Submit button formats node list using JSON and queries servlet
  - Map loads intelligently (i.e. entire 8mb thing doesn’t all load on start)
  - Panning is supported (not necessarily zooming)
  - UI is browser-safe and robust: support resizing, etc.
  - Links to primitive (possibly empty) faq and about
  - Style sheets developed and general site look and feel determined
Full Feature Release

- UI will draw multi-destination routes, where the paths between different destinations will appear in different colors.
- Can handle different path types and transportation types. The UI will allow the choosing of a transportation type (i.e. walk, bike)
- Algorithm will distinguish between different types of nodes (buildings, doors, normal) and will choose routes based on cost.
  - Quick Route Search
  - Lowest Cost Route Search
- Data will be more complete, more nodes, all paths.
- Fully customizable destination list with autocomplete (intelligently suggesting).
- Full support for zooming and panning by tiles.
- Mature, consistent looking interface.
This is a sequence diagram for server-side control flow when a user sends a HTTP request for finding routes for multiple buildings.