Application - Get There

Deliverable 2

You will submit a copy of the Requirements and Analysis Document below to myWPI:

1. **Introduction**
2. **Current system – how are people figuring out how to get around campus now?**
3. **Proposed system**
   1. **Overview**
   2. **Functional requirements**
   3. **Nonfunctional requirements**
   4. **Constraints (“Pseudo requirements”)**
   5. **System models for the complete application to be delivered at the end of the term**
      1. **Product backlog – prioritized list of all user stories (top ones have higher priority)**
   6. **System models for the minimal application to be delivered at the end of the first iteration**
   7. **Scenarios (revised version of project deliverable 1)**
   8. **Use case model (revised version of project deliverable 1)**
   9. **Iteration backlog - user stories (revised version of project deliverable**
   10. **Object model - Jeff**
       1. **Class diagrams for entity and control objects (make sure the diagrams are large enough that I can read them)**
       2. **Data dictionary**
       3. **Boundary objects (user interface) – UI mockups, navigational paths**

* **Form**
* **Map Display**
* **Navigation Display**
  1. **Dynamic models**
     1. **Sequence diagram**
     2. **Statechart diagram (optional)**

1. **Glossary**

Introduction

Get There will be an exclusive navigation application written for the Worcester Polytechnic Institute Campus. The campus is located in the

Current System

Current students are able to find campus maps of the WPI campus in paper handouts and in online 2D and 3D versions. However, while this is helpful for navigating from building to building, it is not helpful when students are trying to navigate to classrooms they have never been too. Many student talk to other classmates and faculty for assistance in finding these classrooms. During the first week of classes, many students struggle to be on time to their lectures and labs. A new solution is needed to help students find their new classrooms faster.

Proposed System

The system our team proposes to produce is a PC software to allow users to obtain directions from one WPI campus location to another, and display the recommended path to follow, overlaid on maps of the campus and of each building, on that user's monitor. Any individual can select a starting location and destination, and our system will determine the shortest walkable route between them and provide instructions for them to use to take it. Features will offer additional information and route options that can be utilized per user preference and convenience.

Functional Features

* Turn by Turn instructions
* ETA/Distance
* Alternative Routes

Export

Map displays important landmarks such as:

* Bathrooms
* Emergency blue towers
* Elevators/Stairs
* Vending machines, food courts
* emergency exits

Access to transportation Schedule

* SNAP
* Gateway Shuttle

Disability access Routes

Search Location of Professors

Multiple destinations

Possible Mobile Application - Android

Non-functional Features

Constraints (“Pseudo requirements”)

* The implementation language must be Java
* The application must work on desktop computers, running Linux, Windows OS, or OS X
* The application must be made available for download online, for free. (Unless instructed otherwise)
* The application must be efficient, and return results as fast as possible

Data Backlog

* As an admission employee, I want to enter paths on the campus map so the application is able to generate routes.
* As a visitor, I want to know how to get to the dining hall.
* As a new student, I want to know how to get to my physics lab.
* As a user, I want turn by turn instructions on how to get where I need to go.
* As a professor, I want know the path distance between Riley and AK for math problem.
* As a student, I want to be able to print the route I plan to take.
* As a student, I want to know how to get to my professor's office.
* As a visitor, I want to know where the nearest bathroom is and how to get there.
* As a freshman, I want to know when the next shuttle is, where it stops, and how to get there.
* As a professor, I want to get a route from the parking lot to the campus center and then to my classroom.
* As a student, if I don’t like my route I want to see another option.

Use case Textual description:

Name: SetRoute

Participating Actor: Application user

Entry Condition:

* User executes the application on a desktop computer
* User enters the origin, i.e. the starting point of the trip
* User enters the destination, i.e. the ending point of the trip

Exit Condition: Passenger sees a map on the screen, with the shortest route from the origin to the destination. Passenger also receives turn-to-turn instructions, with how far to travel during each turn.

Flow of Events:

User inputs origin , and destination, using a keyboard, in the corresponding boxes

User presses the go button

Application generates the map, with a clear path from origin to destination

Application generates turn-to-turn instruction.

Alternate Flow of Events:

InvalidCoordinate:

User inputs invalid origin or invalid destination or both

Requests the User to check the invalid input and re-enter it

CancelRoute:

The User cancels the route selection process and closes the application

* As a developer, I want to be able to upload maps into the map tool so that I can enter nodes and paths.
* As a developer, I want to visually enter nodes onto a map so that the routing application will have reference points.
* As a developer, I want to create edges between nodes so that the program can plan routes between nodes.
* As a developer, I want to create edges between nodes on different maps so that the program can plan routes between maps.
* As a developer, I want to assign distance to edges so the program can find the shortest path.
* As a developer, I want to export the nodes and edges so that I don’t have to hardcode nodes and edges for the path finding program.

Scenario:

Jim is a developer at Pathfinding Studios, and he needs to set up a new map so that the Studio’s proprietary software Get There can find efficient paths on it. He opens up their specialized development tool designed to do exactly that. He uses built in buttons to load an image of the new map. Then Jim clicks on the map in each place a pathfinding node is needed, and a circle is created on those spots to represent the nodes. After he places all of the required nodes, using other buttons to delete and move nodes as necessary, he needs to place paths to connect the nodes. he clicks on the ‘Pathways’ button and begins to click on different nodes. These nodes are then visually connected by lines on the map to represent pathways. Once he is done he clicks another button which creates a new Map class in the Get There program, and the new map is ready to be implemented.

|  |
| --- |
| Interior Building Layouts for route planning |
| Maybe location of bathrooms near destination? |
| Integration into Google Maps |
| Classroom locations/ floors of buildings |
| Population density by time of day, based on where classes are scheduled and when they get out |
| Open parking spaces, current events |
| Push to Google maps on android |
| Mobile version |

|  |
| --- |
| I would imagine that an application on the phone would be the most widely accepted and used. if you could code it on android and is it would get the most possible traffic and use in my opinion. |
| Our campus is so small, we really don't need this |
| only really useful as mobile app |
| a phone app would be way more convenient! |
| A mobile app would be significantly more convenient to use (and thus I would use it more often). |
| this seems pointless? |
| Sounds amazing! |
| Pedometer to track how far you've traveled throughout the day |
| Will not use if it's on the desktop, but might use if it's on mobile. |
| Having your current location based on phone GPS would be helpful |
| This app should also provide snap schedules and be topographically accurate. I want to get around having to go uphill as little as humanly possible |
| I'm actually an alum, I would find the app useful if it told me what's happening on campus and where |
| Now that I think about it, as a student who's been here for a while and already knows their way around campus, this would probably be most helpful to guide visiting friends and family. |
| A mobile app for this would be FANTASTIC, especially for new students and on the first day of classes! |

Textual use cases for map tool:

Name: CreateMap

Participating actor: Developer

Entry condition: Developer executes the map tool

Exit condition: Nodes and edges are exported into a text file

Flow of events:

* Developer presses on upload map, include (uploadMap).
* Developer creates nodes, include (EnterNode) and edges, include(CreateEdge) and assigns edge distances, include(AssignEdgeDistance).
* Developer links a node on one map to a corresponding node on another map, if any, include(LinkMaps).
* Developer exports nodes and edges to a text file, include(ExportNodes&Edges).

Alternative flow of events:

[UndoAction]

* Developer presses on upload map, include (uploadMap).
* Developer creates node, include (EnterNode).
* Developer deletes node by hitting “Undo Action” button.

Alternative flow of events:

[UndoAction]

* Developer presses on upload map, include (uploadMap).
* Developer creates nodes, include (EnterNode) and an edge between them, include(CreateEdge).
* Developer deletes edge by hitting “Undo Action” button.

Alternative flow of events:

[UndoAction]

* Developer presses on upload map, include (uploadMap).
* Developer creates nodes, include (EnterNode) and edges, include(CreateEdge) and assigns edge distances, include(AssignEdgeDistance).
* Developer links a node on one map to a corresponding node on another map, include(LinkMaps).
* Developer deletes the link between maps

Glossary:

Node: A point on a map.

Edge: The connection between two neighboring nodes.

Hardcode: Physically writing code instead of using a program to generate it.