



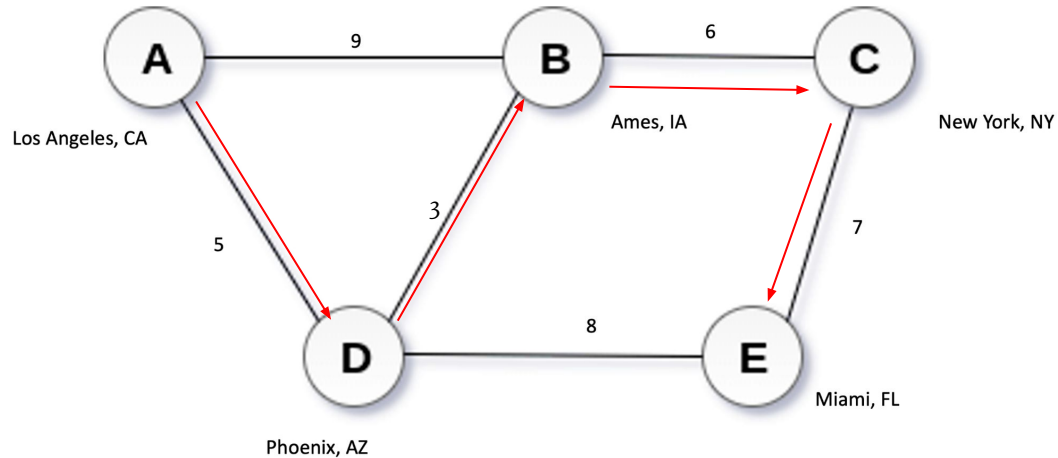
# Trip Plan

Ben Bradley, Cody Nielsen, Eduardo Ramirez, Maxwell Smith, Omar Almehairbi

COM S 402C – Senior Design – Fall 2022

# What problem is being solved?

- Efficient multi-location trip planning
- Turn by turn directions accompanied by street map pictures





# Design challenges/issues, choices, and how addressed?

- Shortest Path Algorithm
  - Brute force was the only viable option
- Getting Real-World location data
  - Resolved by using Open Street Maps & getting API key
- Using Street Map Pictures
  - Resolved by using Google Maps & getting API key
- Using a No-SQL locally with port forward issues & IP whitelisting issues
  - Used MongoDB cloud, whitelisted our IPs
- State management issues at demo 2
  - Re-did this area with correct React state management
- CI/CD Issues
  - Removed Firewall to open port, Gitlab runner had issues everytime VM restarted
- Not enough time for all user roles
  - Focused on logged in users and then guests, did not prioritize admin

# Business Canvas Model



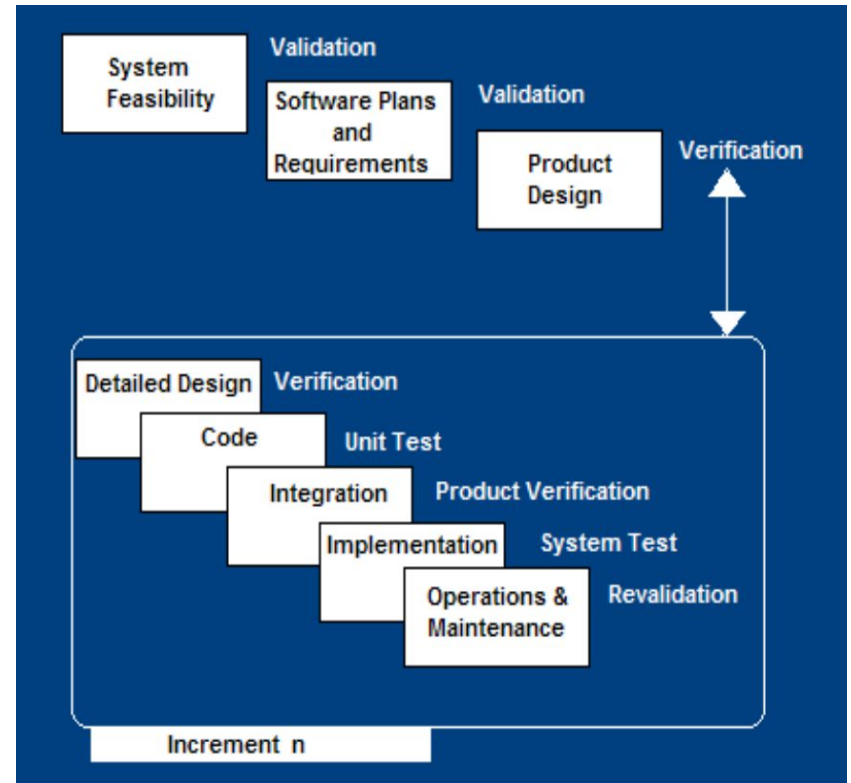
<p><b>KEY PARTNERS</b></p> <ol style="list-style-type: none"> <li>Travelers</li> <li>Hotels and restaurants</li> <li>Rental car agencies</li> <li>Resorts</li> <li>Tour operators</li> </ol>	<p><b>KEY ACTIVITIES</b></p> <ol style="list-style-type: none"> <li>Sales</li> <li>Service operation</li> <li>Marketing</li> <li>Compare service from other trip planning apps</li> </ol> <p><b>KEY RESOURCES</b></p> <ol style="list-style-type: none"> <li>Human resources</li> <li>Technology</li> </ol>	<p><b>VALUE PROPOSITIONS</b></p> <ol style="list-style-type: none"> <li>Providing <b><u>EFFICIENT</u></b> trip planning services</li> <li>Hotel booking</li> <li>Rental car booking</li> </ol>	<p><b>CUSTOMER RELATIONSHIPS</b></p> <ol style="list-style-type: none"> <li>Direct communication</li> <li>Live chat</li> <li>E-mail</li> </ol> <p><b>CHANNELS</b></p> <ol style="list-style-type: none"> <li>Website</li> <li>Hotline</li> <li>Mobile</li> <li>Social media</li> </ol>	<p><b>CUSTOMER SEGMENTS</b></p> <ol style="list-style-type: none"> <li>All-inclusive market</li> <li>Family vacations</li> <li>Students</li> <li>Young couples</li> <li>Travelers</li> <li>Company</li> </ol>
<p><b>COST</b></p> <ol style="list-style-type: none"> <li>Human resources expenses</li> <li>Assets (Computers and IT Systems)</li> <li>Rent and utilities</li> <li>Marketing Campaigns</li> </ol>			<p><b>REVENUE STREAMS</b></p> <ol style="list-style-type: none"> <li>Commission direct</li> <li>Fee transactions</li> </ol>	

# Software development practices and tools used

## Software Development Lifecycle

### [Incremental Model]:

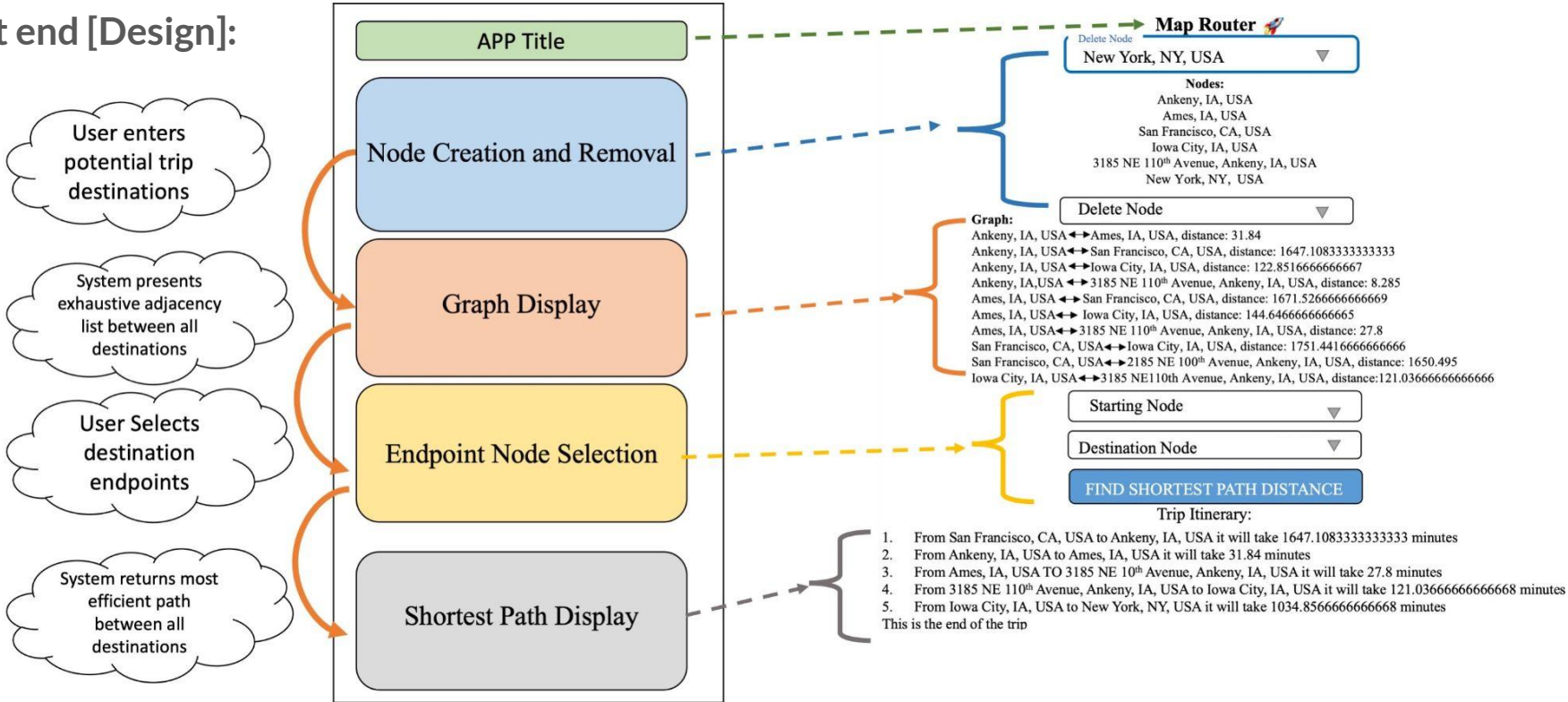
- **Functional Requirements Known Up Front**
  - User Inputs Destination
  - System Generates Path
- **Increased Functionality Across Several Iterations**
  - Node Deletion
  - Turn-by-Turn Directions [with image support]
  - Login
  - Saved States
- **Stakeholder Input at Every Release**
  - Sponsor [Mr. Shetty] engaged in multiple meetings with the group evaluating progress



From: Bruegge & Dutoit Object-Oriented Software Engineering

# Software development practices and tools used

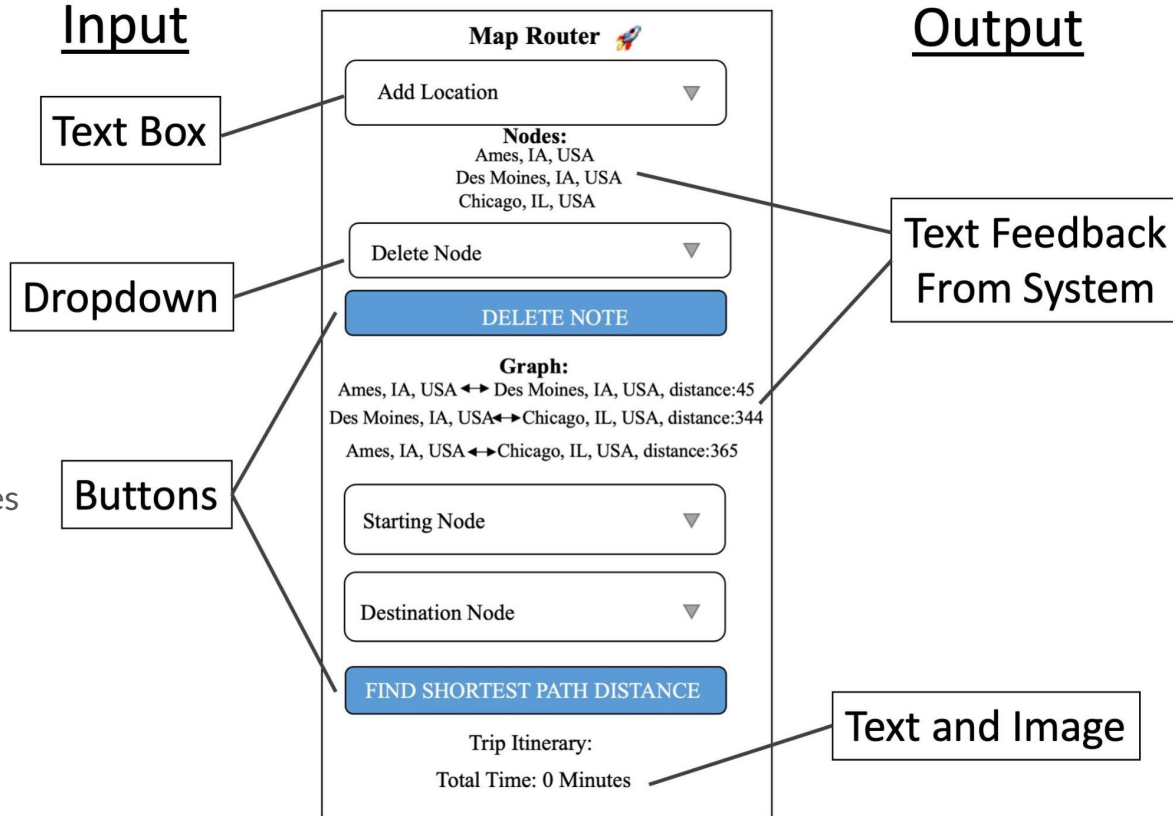
## Front end [Design]:



# Software development practices and tools used

Front end [Design]:

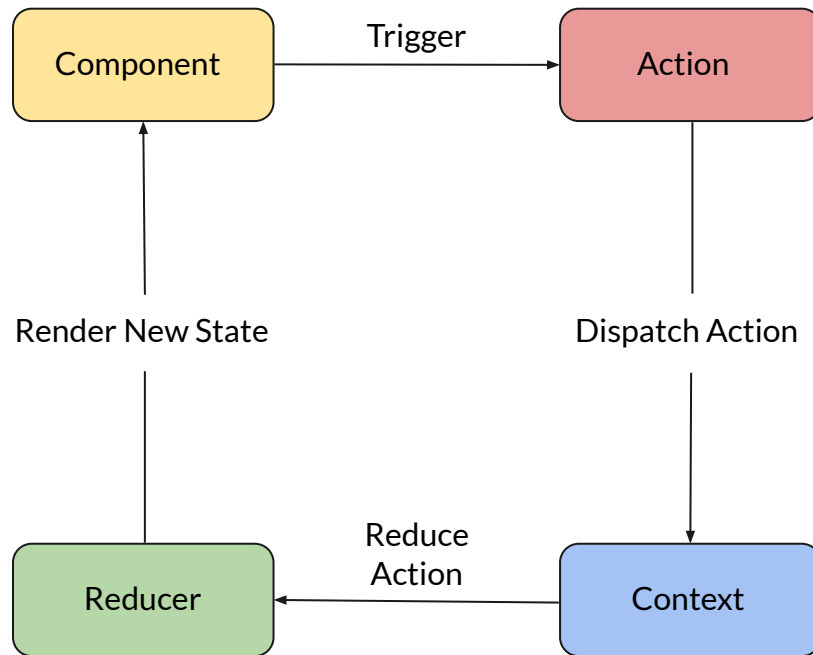
- Segmentation of input and output elements
  - Destination [nodes] inputted through textbox
    - Constrained superset of endpoints through *autocomplete* via API call
  - Subset of chosen destinations grouped in dropdown/combo box
  - Event handling with selected nodes occurs with button-pressing
  - System shows textual graph of nodes as adjacency list
  - System presents and summarizes optimal path with text and images



# Software development practices and tools used

## Front end [Technologies]:

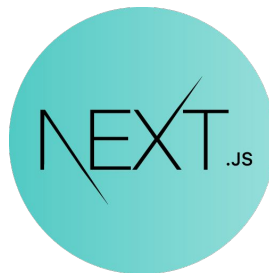
- **Nextjs Pages**
  - Build React-Native components, put them into pages
  - Route between pages using Nextjs router
- **Context and Reducer state management**
  - App context wraps components and gives them a reference to the state
  - Instance of a graph is stored within the app context
  - Reducer contains defined actions
    - Actions interact with and modify the graph
    - User triggers action
    - Component dispatches action to the reducer



# Software development practices and tools used

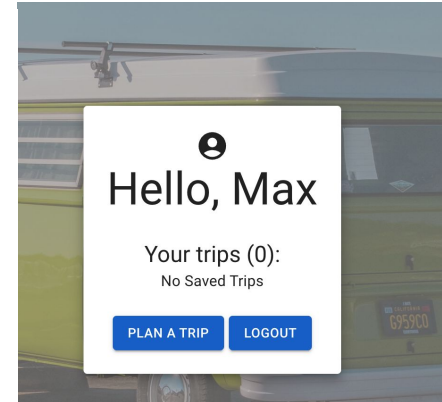
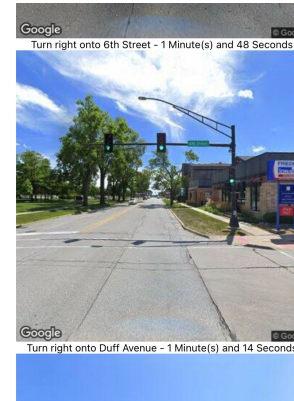
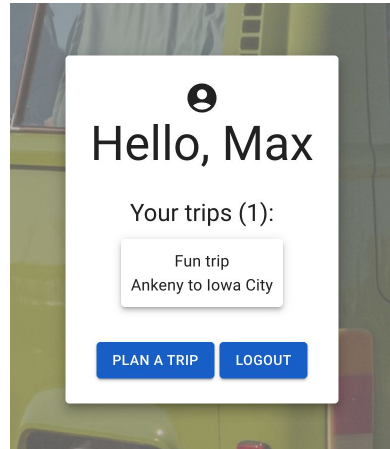
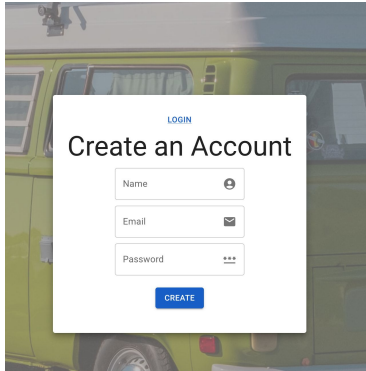
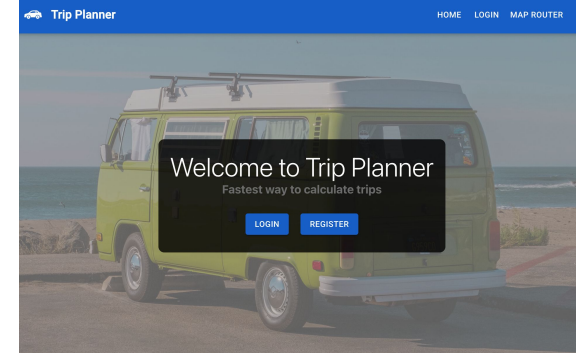
## Back end:

- **MongoDb**
  - “Tables” are called “collections”
  - Stores JSON objects as “documents”
  - Create schema from interface
  - Establish connection as Mongo client object
  - `.find()` `.insert()` // etc
  - MongoDB TripCluster viewer
- **Post example**
  - Method on controller gets called
  - Establish database connection
  - Compile a new model and send it to Mongo
  - Mongo handles the request and posts data to a collection
- **Overview**
  - User controller
  - Front end indicates that it needs data
  - Provide an API endpoint with the necessary request information
  - Node creates and sends the request



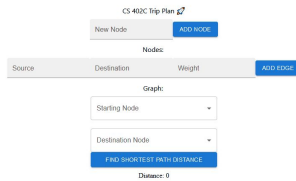
# Final Product Overview

- Trip Planner is a fully featured multi-location trip planner
- Turn-By-Turn Directions with street view pictures
- Saving Routes
- User login/Signup



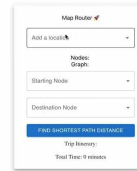
# Progress from demo to demo

## Demo 1 - Week 5



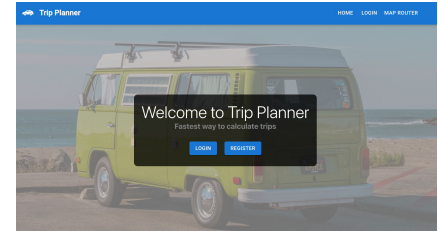
- Setting up requirements
- Setup of Project
- Graph Classes

## Demo 2 - Week 10



- Implementation of real-world map data
- MongoDB Setup
- Login/Register User
- Fixes to state management
- Better UI Design

## Demo 3 - Final



- User Profile
- Saved Routes
- Home Page
- Nav Bar
- Google Street Pictures
- Turn By Turn Directions



## Team Challenges encountered.

- Working/meeting remotely
- Client was in India at the beginning of the semester, lots of implementation freedom
- Learning how to use development tools used & software



# Summary of Contributions

- Ben
  - MongoDB Setup
  - Login and Account registration services
  - Login/Registration/Profile UI
  - Setup app state management
  - API endpoints and request handlers
- Max
  - Home Page + Nav Bar UI
  - API Endpoint setups
  - Integration of 3rd party API services for Mapping data + Lookup component
  - Street Map photos on directions
  - CI/CD Setup
- Eduardo
  - Dijkstra's Algorithm
  - Shortest Path permutations
  - Trip Itinerary
  - Delete Node function
  - Save trip UI
- Cody
  - Docker instance of Mongo
  - Turn by Turn directions
- Omar
  - Time output formatting [for graph and shortest path]
  - Website Maintenance
  - Screen Sketches / UI Planning
  - Final Report
  - UML Modeling
  - Business Canvas
  - Meeting Logs



## Mobile Demo + Live Demo

