OTP MOD Sandbox: Technical Overview

Wednesday, January 18, 2017

Overview

- Review Conveyal Scope of Work Requirements
- Review Existing OTP UI Resources
- Why a New UI Framework? React/Redux Basics
- Proposed Project Structure and Development Milestones
- Technical Questions to Be Addressed
- Preliminary Design/UX Concepts

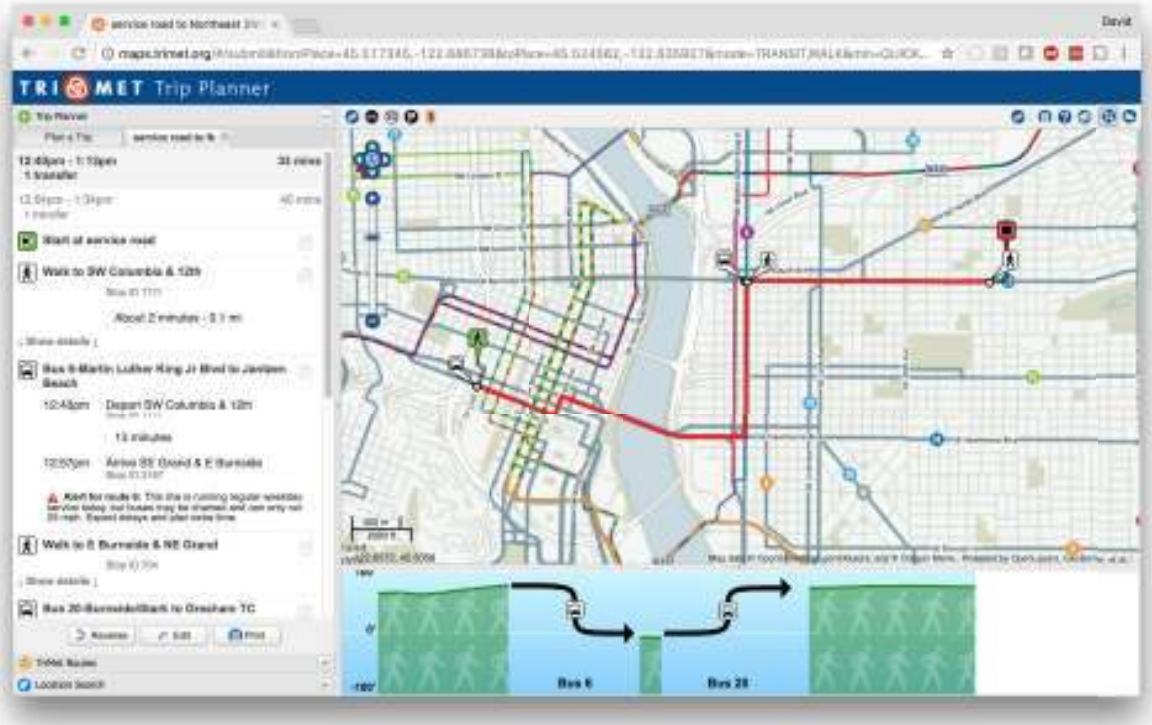
Key Requirements of Conveyal Scope of Work

- OpenTripPlanner Routing/Back-End Enhancements
  - Extend OTP to Support Shared-Use Mobility Services
  - Improve Support for Real-time Transit Information
  - Improve Support for Pedestrian and Wheelchair Routing
  - Improve Bikeshare Support including GBFS Import

Key Requirements of Conveyal Scope of Work

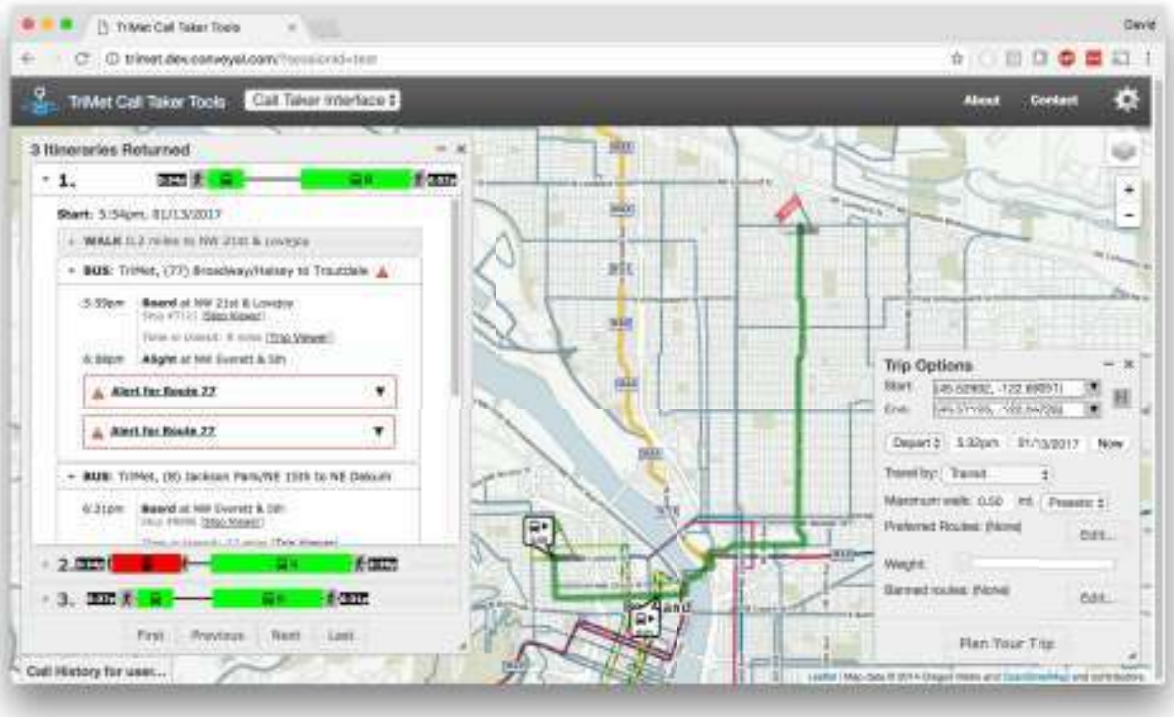
- Create Comprehensive New UI Framework
  - Fully Responsive UI For Use on Desktops and Full Range of Mobile Devices
  - Use React and Redux Architecture
  - Support Itinerary- and Profile-based Routing
  - Integrate with Pelias Geocoder
  - Incorporation of OTP Routing Enhancements (e.g. SUM Support) into Narrative/Map Display

Existing UI frameworks: TriMet Current



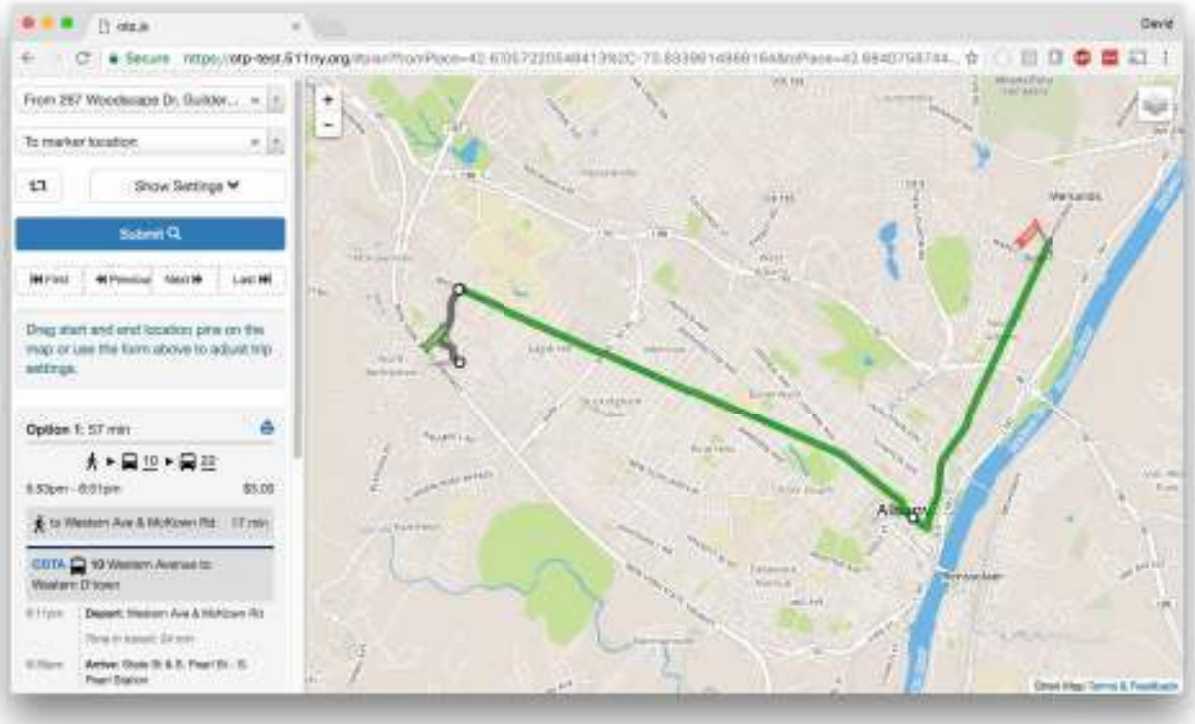
- Circa 2009-2010
- Written in ExtJS and OpenLayers
- Not Responsive

Existing UI frameworks: OTP Default



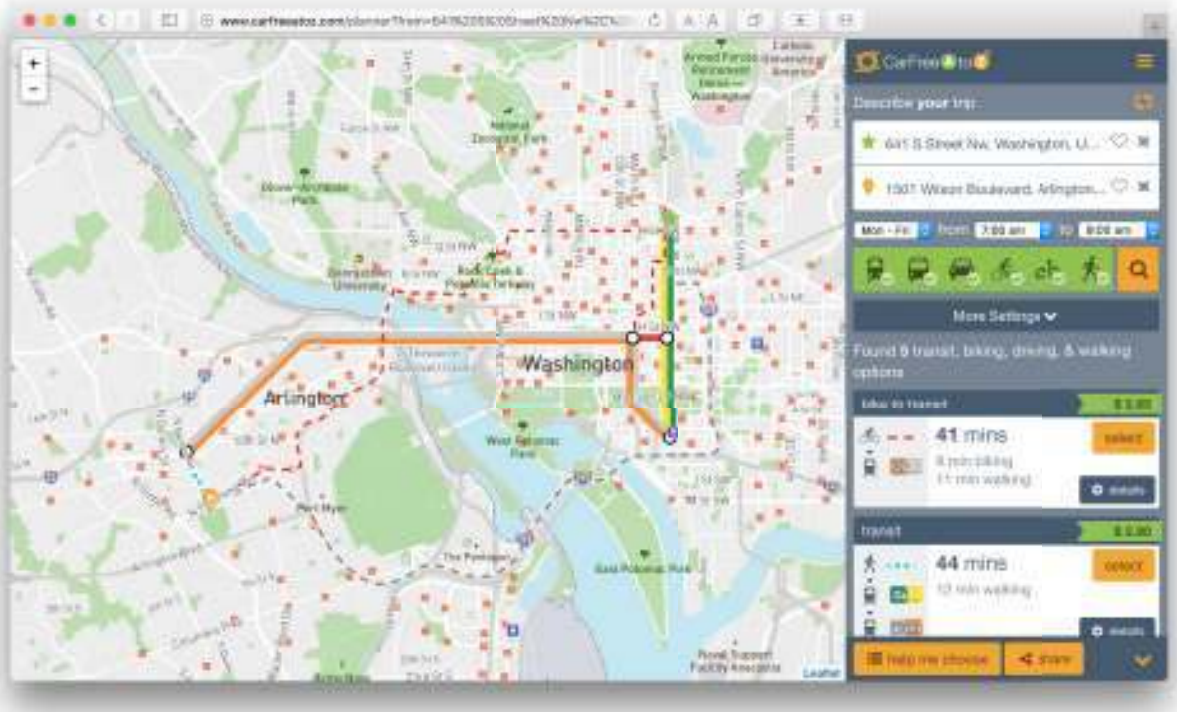
- Circa 2012
- Written in JQuery UI and Leaflet
- Not Responsive
- Includes TriMet Call-taker tools (still in use)
- No dependency management / build system to speak of
- Monolithic codebase; difficult to maintain

Existing UI Frameworks: otp.js



- Circa 2014; developed as lightweight, responsive alternative to stock OTP UI
- Pluggable Geocoding Support
- Uses backbone.js MVP framework
- Uses component for build management (deprecated)

Existing UI Frameworks: Modeify



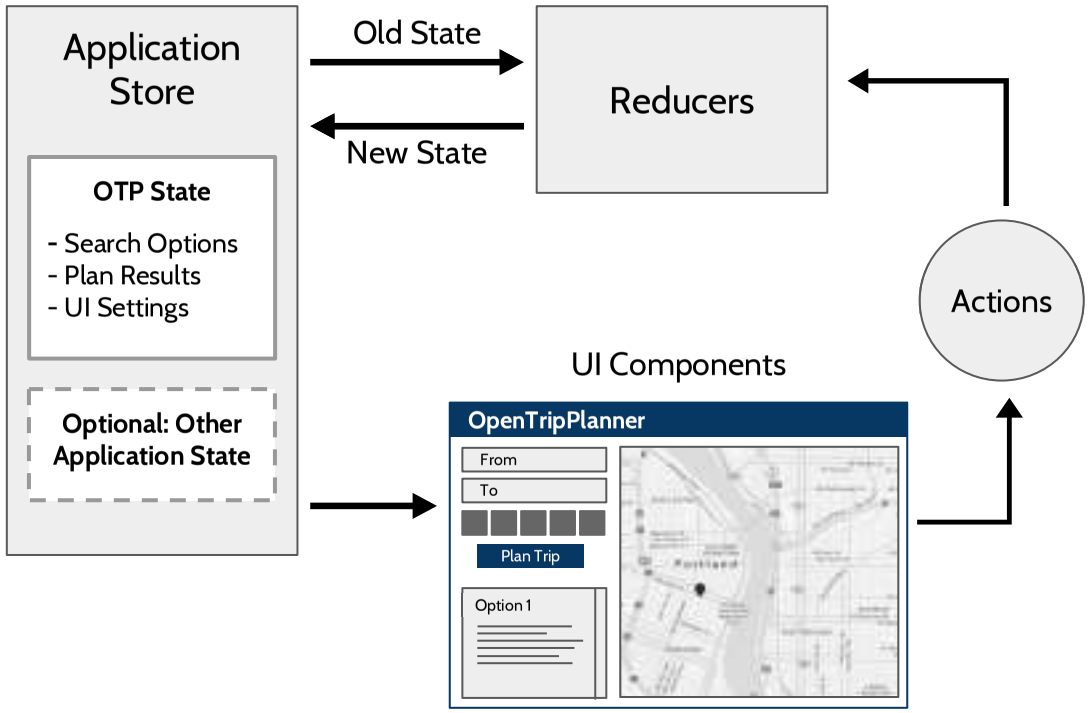
- Circa 2014, developed for Arlington County (Va.) TDM Agency
- Focus on “profile-based” routing
- Uses component for modeling/templating (deprecated), internal Conveyal tooling for build management

Why Another OTP UI Framework?

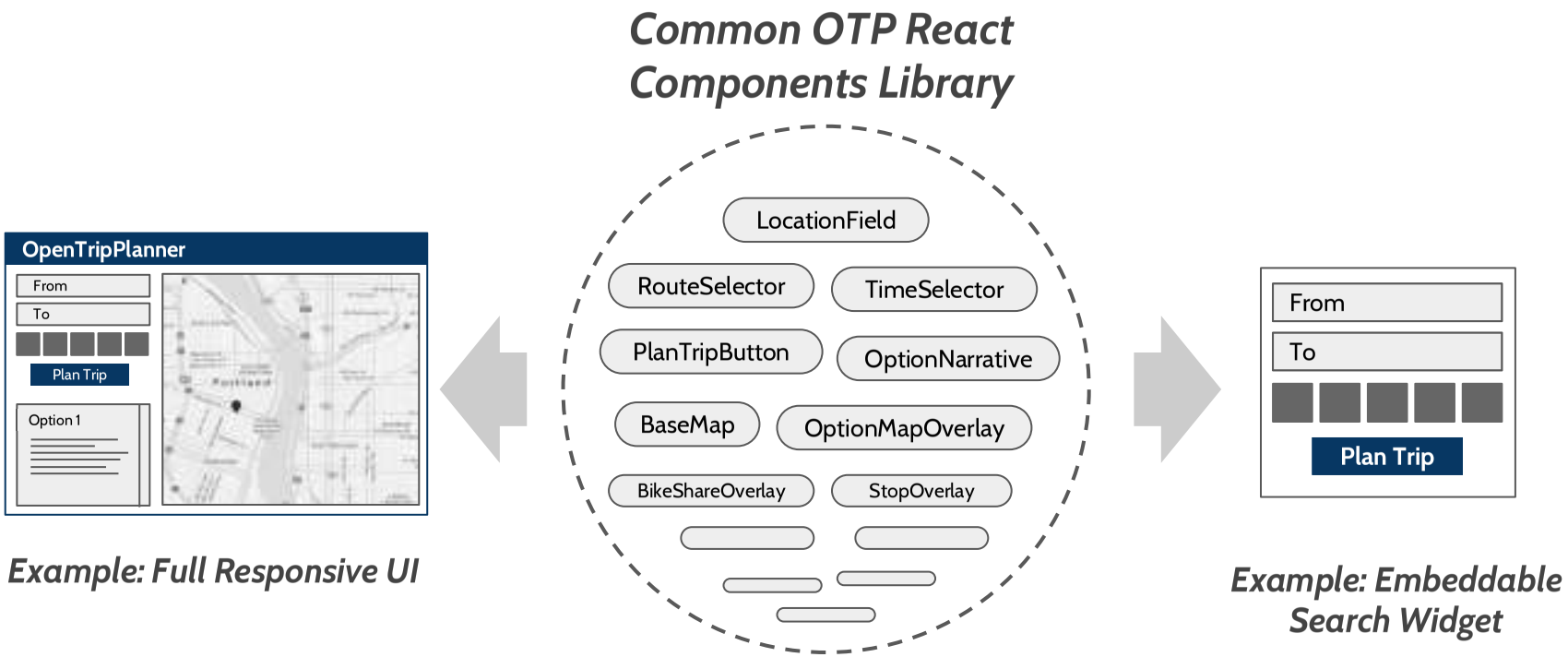
- Javascript Ecosystem has Evolved Rapidly
- New Frameworks like React Help Enable Development of Truly Modular, Extensible Library
- Opportunity to Combine Best Aspects of Existing UIs (e.g. Route Viewer from TriMet UI, Profile-based Display from Modeify)

React/Redux Basics and Architecture

- Central Store includes Entire Application State - “Single Source of Truth”
- One-Way Flow Of Information From State to UI
- State is Transformed via Actions and Pure “Reducer” Functions
- Separation of Presentational (“Dumb”) and Container (“Smart”) Components

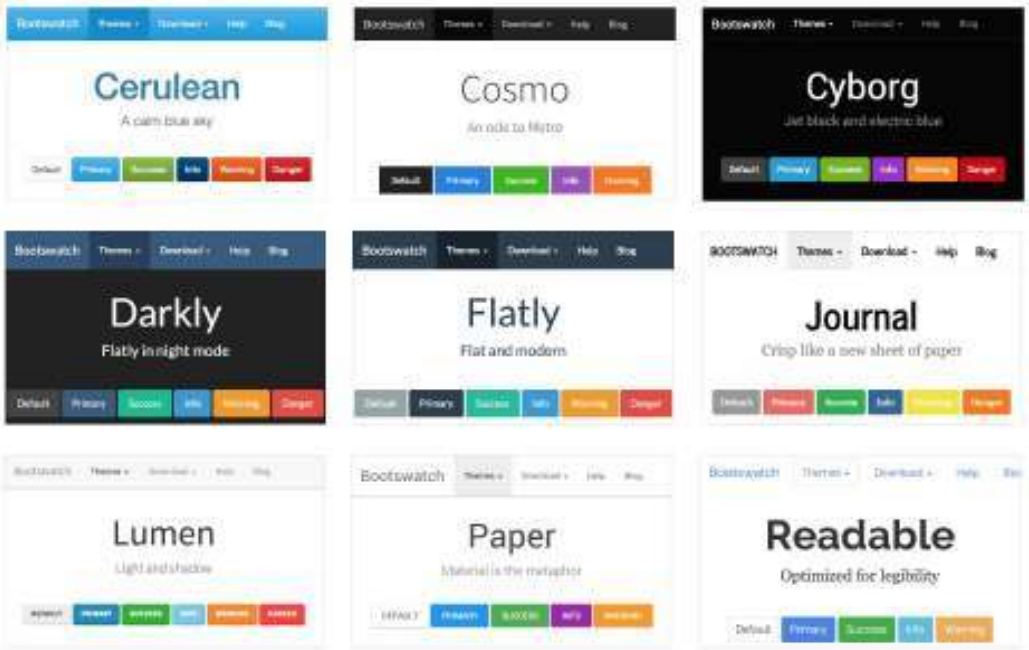


Reusability of UI Components



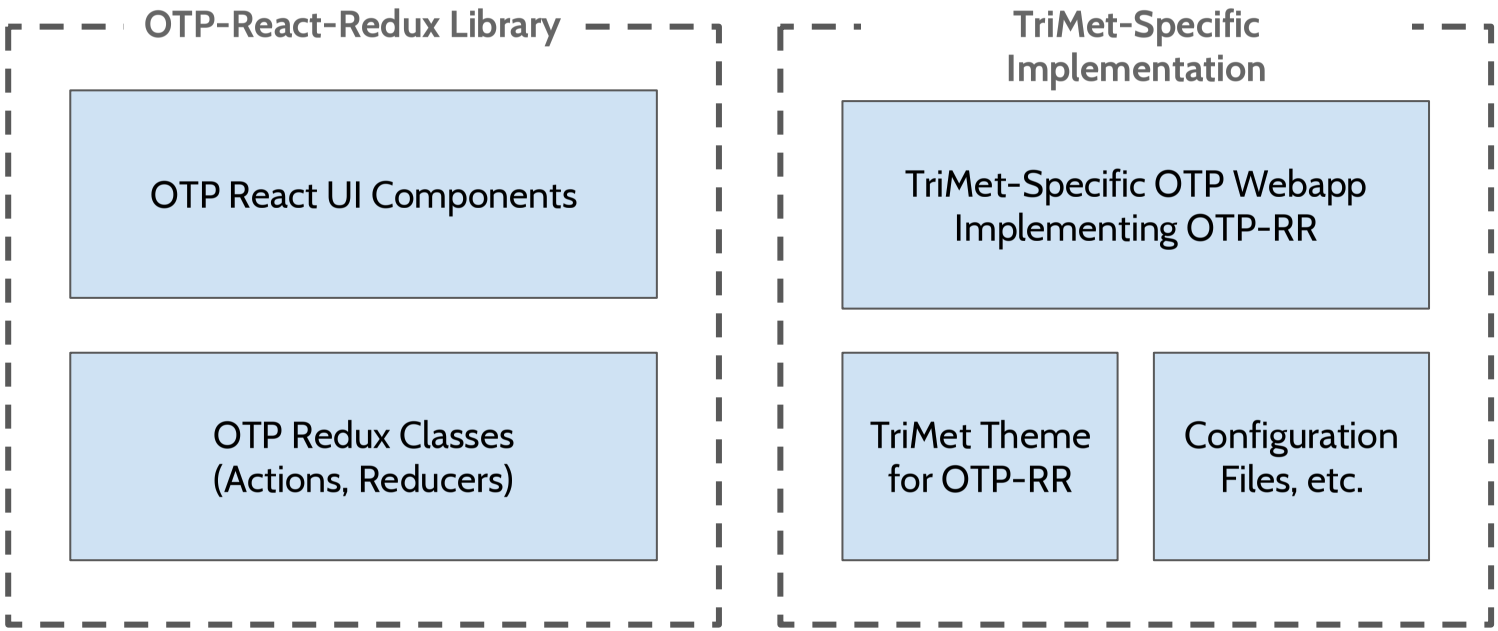
UI Styling

- Look and Feel of UI Should be Separated from Core Component Definition
- Consider Supporting CSS-based “Themes” for OTP React Components
- Option to Use Established Base UI Framework with Existing Theme Ecosystem (e.g. Bootstrap)



Example of Bootstrap themes from bootswatch.com

What We Propose Building



Proposed Development Milestones

- **Milestone 1**: Itinerary-Based Trip Planning
- **Milestone 2**: Geocoding, Bikeshare Support, Profile-Based Trip Planning
- **Milestone 3**: Real-time Integration, Advanced Mapping
- **Milestone 4**: Shared-use Mobility, Wheelchair/Ped Routing
- **Milestone 5**: Extended UI Functionality

Key Technical Considerations

- How will new OTP UI be Integrated w/ TriMet Website?
- What Build/Dependency Management System to Use?
- Build UI Components using Existing Framework such as Bootstrap?
- What Future Extensions are Anticipated? (e.g. GTFS-Flex Support) How do these Impact Application Design?

Design & UX Topics for Tomorrow

- Define System Use Cases
  - High Level Features and Flowchart
- User Goals for Routing
  - RT, Itinerary vs. Profile Based
- Establish Focused User Testing Process
  - Metrics of Success & Testing Team
- Brainstorm Session User Journeys & Scenarios
- Conduct Initial Guerilla Testing Session
  - Handmade Experience Prototypes
- Discuss Mode Filtering
  - TNCs from a User Standpoint, Dealing with a Proliferation of Options

ASSUMPTIONS

- Either tourist or resident with a beginner level of experience with TriMet transportation
- Owns a smart phone
- All ages

1. OPTIMIZE DEVICE EXPERIENCE

- DESKTOP
- MOBILE

2. PRIMARY USE CASE

- TRIP PLANNING

3. USER SEARCH INPUTS

A. ADDRESS SEARCH

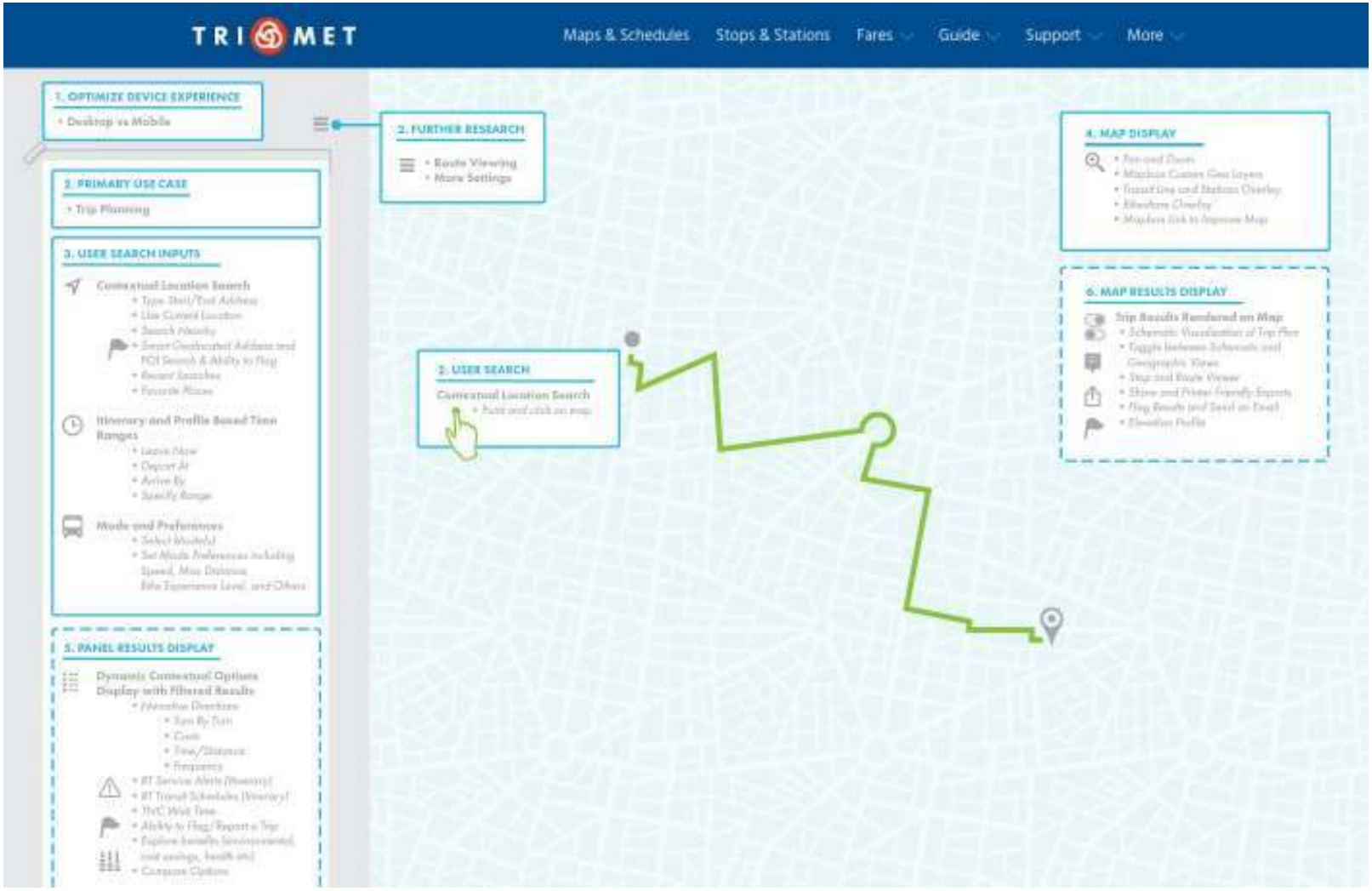
- Type Start/End Location
- Smart Geolocation
- Use Current Location
- Flag Results
- Search Nearby
  - Search Stations Nearby
  - Search/View POI Nearby

Option A

ITINERARY SEARCH

Option B

ITINERARY SEARCH


TRIMET
Maps & Schedules
Stops & Stations
Fares
Guide
Support
More
1. OPTIMIZE DEVICE EXPERIENCE
• Desktop vs Mobile
2. FURTHER RESEARCH
• Route Viewing
• More Settings
2. PRIMARY USE CASE
• Trip Planning
3. USER SEARCH INPUTS
Contextual Location Search
Itinerary and Profile Based Time Ranges
Mode and Preferences
5. PANEL RESULTS DISPLAY
Dynamic Contextual Options Display with Filtered Results
4. MAP DISPLAY
6. MAP RESULTS DISPLAY
Trip Results Rendered on Map
1. USER SEARCH
Contextual Location Search