

Crowd Journalism Platform

Rouke Pouw 15-07-2019

Contents

- **One liner**
- **Key Design Considerations**
- **Wireframe Conceptual Design**
- **Data Model**
- **Architecture**

One liner

Online collaboration platform to collect and asses sources, establish facts and present timeline visualizations.

Key Design Considerations

- Modern UI
 - o utilize direct input methods:
 - refrain from save buttons but save data directly
 - drag and drop
 - o Reduce context switching
 - o Reduce clicks
 - o Natural text interface instead of itemized data entry.
 - o Multi User Environment
 - Seamless real time cooperation using state subscriptions
- Minimized Functionality Methodolgy
 - o Only provide functionality that is clear to large majority of users instead of catering to power users.
- Strict Separation of Concerns
 - o Separate UI from API
 - o Modularize back end
 - o Abstract Data Model : separate business logic from data model
- Single source of truth
 - o use 'one stop shop' parameter files instead of definitions spread out over multiple files.

Wireframe Conceptual Design

Login

Username	<...>
Password	<...>
<input checked="" type="checkbox"/> Remember me	
[Login]	
[Create New Account]	

Welcome

\$Account.Name	[My account]	[log out]
Select Project		
- [List projects]		
[New Project]		

Workspace

\$Project.Name	[Switch Project]	\$Account.Name	[My account]	[log out]
----------------	----------------------------------	----------------	------------------------------	---------------------------

Toolbar [Toggle SourceNavigator] [Toggle AccountNavigator] [Toggle FactNavigator] [Toggle NavigatorDrillDown]	
Navigators	NavigatorDrillDown

Toolbar

Toggle SourceNavigator : Show/Hide list of available sources.

Toggle FactNavigator : Show/Hide time line of facts

Toggle AccountNavigator : Show/Hide list of accounts

Toggle NavigatorDrillDown: Show/Hide list the entity details for the currently selected

Navigators

The navigators provide list overviews with filter options for the given entities. Their horizontal size can be modified by dragging their the vertical line separating them.

SourceNavigator will provide a list of sources.

- Input options
 - Quick toggle to add/remove source from project
 - Multi select tools to manage labels and toggle to add/remove source from project
 - Sources can be dragged to facts to add sources to facts
- Filter options:
 - Added to project (true/false)
 - text
 - labels
 - period
- Drill down
 - Clicking on a source will open the source details and input options in the NavigatorDrillDown

FactNavigator will provide a chronological ordered list of facts.

- Input options:
 - Text Mode
 - a single line input for Title.
 - a date pick input for period
 - a list of references sources
 - an indicator if no sources are provided)

- sources can be drag & dropped here
 - clicking on a source will drill down on that source
 - Visualization Mode
 - Horizontal / Vertical Timeline
 - Show Year/Month labels
 - Format options based on Labels
 - More.. (Future extensions)
- Filter options
 - Text
 - Labels
 - Period

AccountNavigator will provide a list of accounts

- Input options
 - (group)admin : Multi select tools to add/remove account from project
- Filter options
 - Added to project (true/false)
 - text
 - labels
 - project
- Drill down
 - Clicking on an account will open the account details and input options in the NavigatorDrillDown
 -

NavigatorDrillDown

The NavigatorDrillDown will provide entity details and input options for the current selected entity. It will always appear on the rightmost side.

Data Model

- Entity
 - id
 - title
- Label : Entity
 - Color
- Group : Label
- Account
 - Email
 - username
 - password(hash)
 - Access: Group -> (Member/GroupAdmin)

- Source : Entity
 - file
 - properties{}
 - labels[]
 - permissions: Group -> (Read/Write/Remove)
- Project : Label
 - permissions: Group -> (Read/Write/Remove)
- Fact : Entity
 - date/period
 - project
 - sources[]
 - labels []

Architecture

- Server
 - Router
 - Serve front end HTML/CSS/JS files to client
 - Handle API requests from client
 - REST API based on entity queries
 - Data Handler
 - Validate requests with defined data model
 - Subscription Handler
 - Handle session login/logout
 - Communicate state changes to subscribed entities
 - Storage Handler
 - Use abstract storage interface
 - Store source files
 - Store entity data