NoteWorks – A Better Way to Take Notes
By: Liam Doran (2014), Advisor: Professor Ridgway
Trinity College Computer Science Department

**Elements**
- **Node**: Topic, focus of attention
- **Note**: Brief, single piece of information
- **Link**: Connection between two nodes
- **Category**: Marker of similar nodes

**Goals:**
- Parse-able: Comprehensible to backend
- Simple: New users can learn quickly
- Robust: Organizational options for user

**Node**: Title
**Node & Category**: Category Name, Title
**Notes**: - Content
**Links**: :Title, Title, Title, etc.

**Introduction**
**Objective**
Build a web application that uses graph drawing and a simple interface to help users create cleaner, better organized, and more usable notes.

**Motivation**
Studies have shown that notes taken on a computer with organization in mind result in better learning. But, most notes today are barely organized and physical, making them hard to maintain and study from.

**Approach/Design**
The application works by allowing a user to type in notes with a simple markup language I designed. The app parses these notes, breaks them down on the backend, and on the frontend it presents the user with an interactive graph of their data that updates in real time as they type their notes.

Noteworks was built with Ruby on Rails, Cytoscape.js, and Foundation, as well as JavaScript, Ruby, Hami, and SASS.

**Data Flow**

**Initialization**
- **Backend (Ruby)**
  - Retrieves notes from database.
- **Frontend (JavaScript)**
  - Receives JSON, initiates graph drawing.

**Live Updating**
- **Backend (Ruby)**
  - Watches text for changes in caret position, number of lines, selection size, and content.
  - Database (data and relations) updated
  - Changes tracked (ARM), converted to JSON
- **Frontend (JavaScript)**
  - AJAX response captured, graph updated with submitted JSON.

**Graphing**

**Algorithm**
Uses a Force-Directed Graph Drawing Algorithm
- Models the graph as a physical system
- Nodes repel each other
- Edges are springs, resist compression and stretching

The graph iterates until it reaches an equilibrium, where the benefits of further changes fall below a threshold.

**Random Start**
- B-G pull close
- R-G, R&BG repel

**Iterations**
- B-G, BBG repel
- R-G pull close

**Equilibrium**

**References**
