

# Codename: gg

Project Name: Gossip Guy

Team Members: Cory Eurom, Xing Xu, Katie Ho

Website: <http://staff.washington.edu/xingxu/gossipguy/>

## Report

### Introduction

Gossip Guy was created as the final project for the class INFO 445 Spring 2013 Advanced Relational Database Design, Management, and Maintenance. The project uses tools and technologies available at the University of Washington.

### Application Area/Framing

The Gossip Guy application has the basic features that would be applicable to an online tabloid website. Reporters are able to write gossip about celebrities. Gossip Guy will allow users to add and update reporters and celebrities. In addition, gossips can have tags, which are a subset of bundles. Both bundle and tags would help with the organization of gossips and with navigation on the website. An example of a bundle might be couples, and an example of tags might be 'Brangelina.'

Users would be able to change the status of the gossip by moving it along a workflow. Because the process for online publishing is complex, the Gossip Guy application allows for flexibility in the process. Thus, as the organizations who use the Gossip Guy application grow, their process can also change. For example, they may need to add a new stage in the process as such as 'factcheck' or 'legal review.'

### Requirements

We have implemented all necessary back-end functions such that the Gossip Guy application can perform the basic tasks such as add, delete and update. The functions will be called by a command prompt that follows strict syntax and command structure. The commands are identified by key words and must be followed by the correct amount and type of parameters, otherwise an error will be reported.

### Performance and Scaling

The only area in which concern would arise with regards to scaling is as time progresses, there would be a need to remove workflows and gossip. Celebrities are not of much concern because only a limited number of Celebrities exist and much more data will be going into the gossip related tables. Tag and bundle entries will rely mostly on the growth of gossip. Gossip and its bridge tables will be the most commonly updated tables because they make up the core of our application. The main function will be to track gossip so this table will be updated and queried on a regular basis depending on the amount of users.

## Strength and Weaknesses

### Strength

We allow for multiple reporters who can report on multiple celebrities.

### Weaknesses

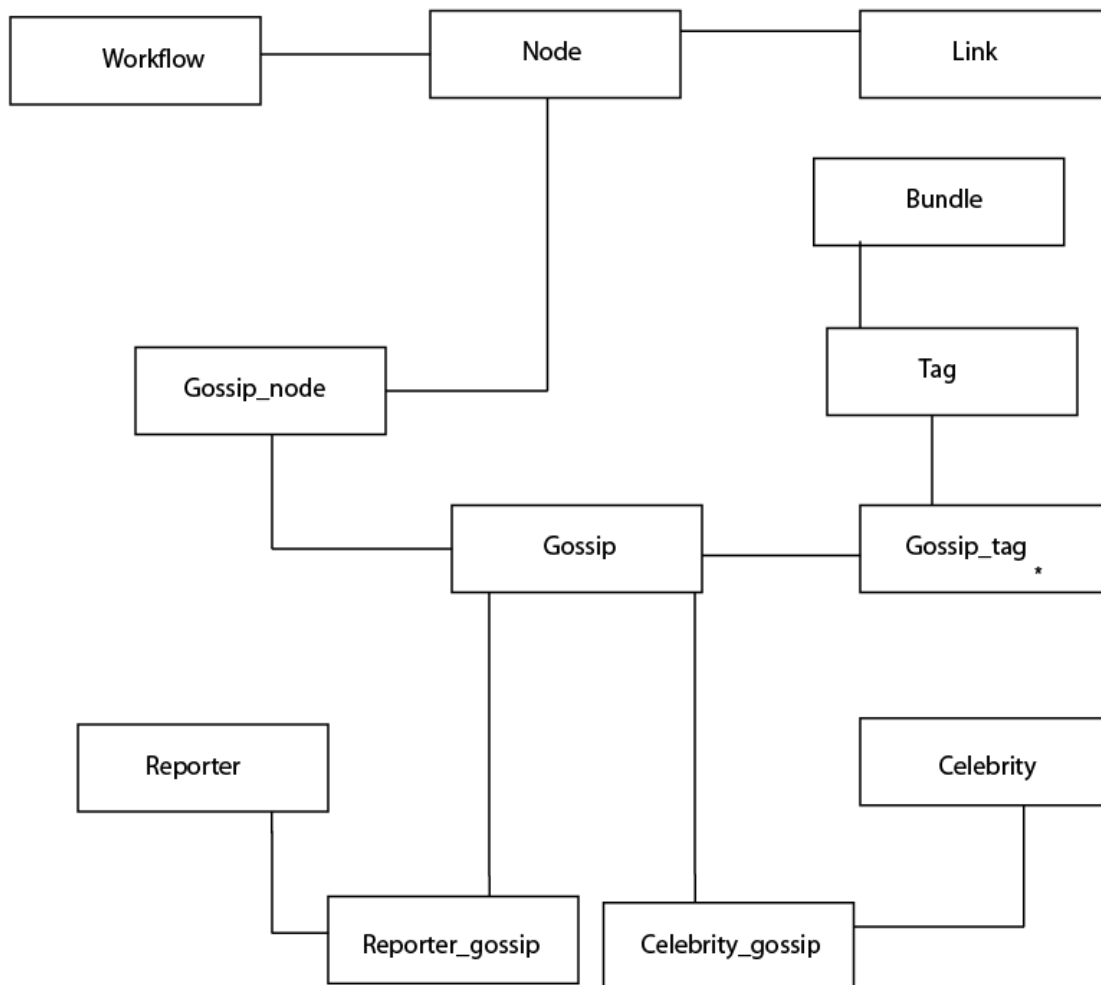
Reporters cannot revert to earlier drafts.

There is no security. Reporters have no password.

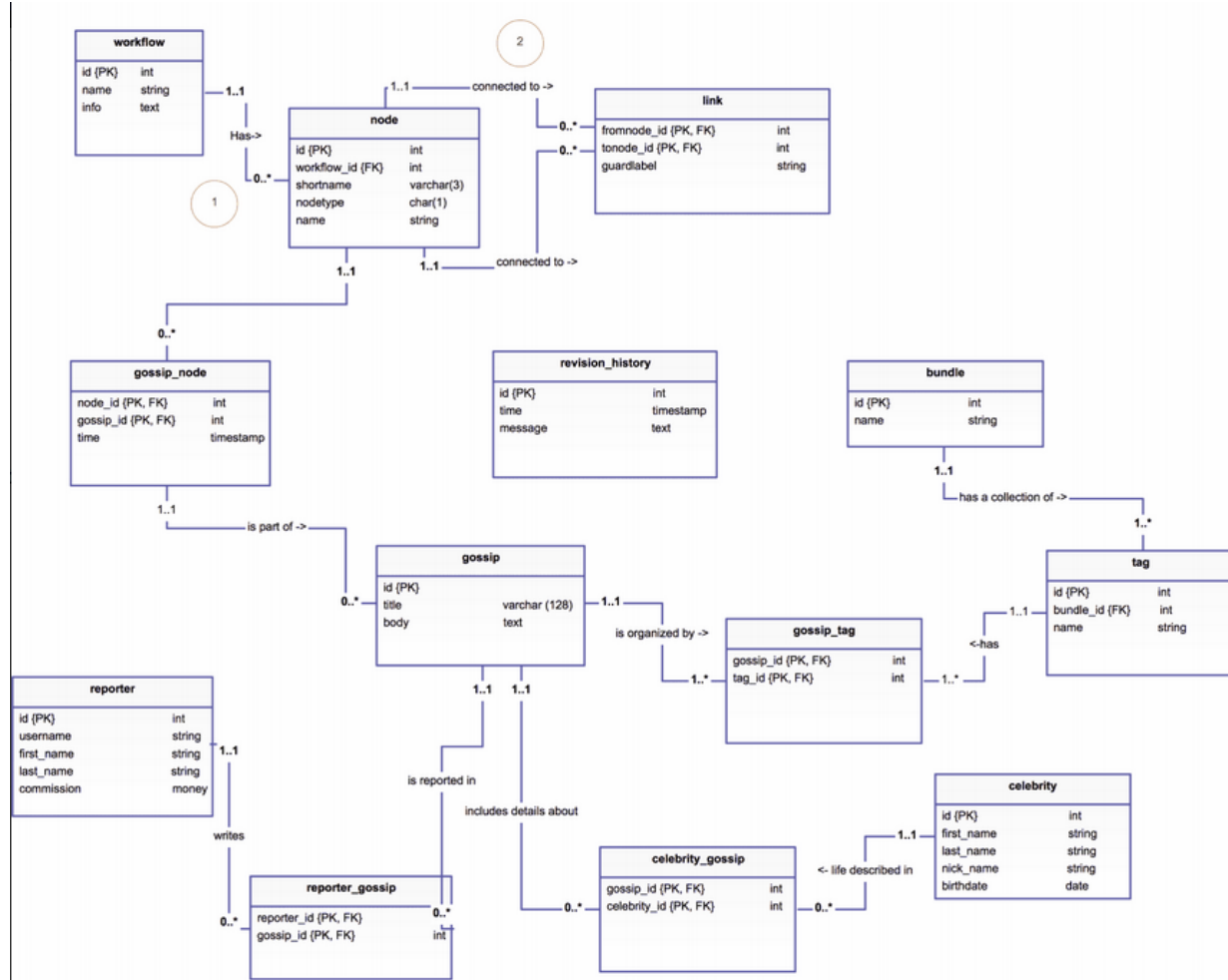
All users currently need to use the command shell.

The terminal's error messages are not informative.

## Conceptual Data Model



## Logical Data Model



### Workflow Module

1. The model allows for the creation of multiple workflows. One workflow can have zero or many nodes. Nodes belong to only one workflow.

2. A node can relate to another node through a link. There can be zero or many links for each node. Each link connects only two nodes together. Link has foreign keys fromnode\_id and tonode\_id that matches with node.id, the primary key for Node. Fromnode\_id and Tonode\_id are composite primary keys. Each pair of fromnode\_id and tonode\_id is unique.

### Documentation Module

A Node can have many gossip.

Many Reporters can work together to write Gossip.

Gossip can be about multiple celebrities.

### Tagging Module

Each tag has only one bundle. A bundle can have many tags.

Each gossip can have many tags.

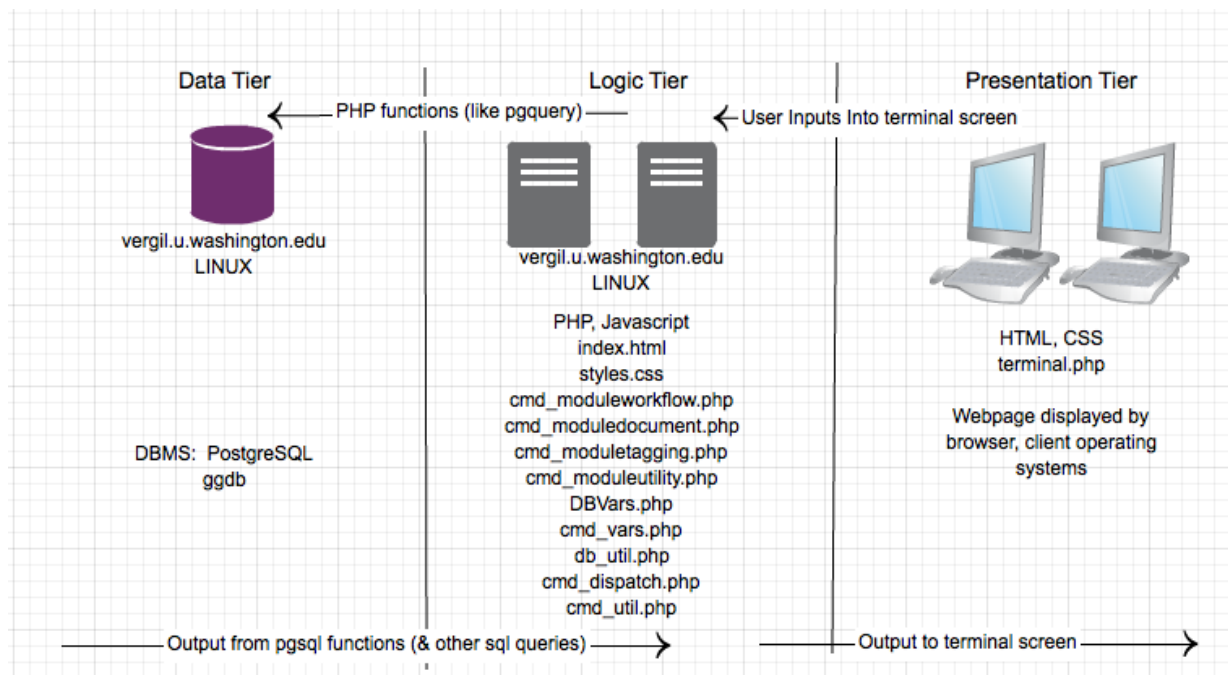
### Utility Module

The table revision\_history will allow technical super users to keep track of key additions, modifications, and changes in the Gossip Guy system.


## Physical Database Model

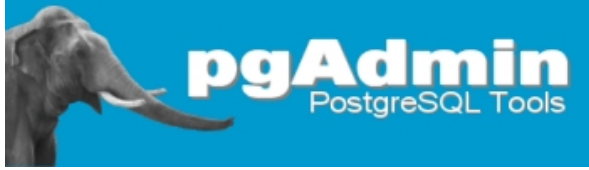

To be filled after draft.

## System Architecture Diagram



## Summary of development tools and processes

	Git and Github was used for source control
---	--

	pgAdmin used for postgresSQL development
	Text editors Sublime Text 2 and Notepad++ for PHP, HTML, and CSS editing.

The Gossip Guy project team separated most of the coding work according to functions. Thus, the people who created the backend function in pgsq also write the PHP function that calls it.

Each team member had their own student web server, which they uploaded synced copies of the Gossip Guy application, differing only in DBVars.php, which holds the connection details to the specific server. Thus, the team relied heavily on github for collaboration.

## Appendix A: Reflections

We have a super strong group though we were small and grouped together haphazardly. Our process is the best. Everyone in the class will be super impressed.