

Whatever: Elizabeth Paperno, Anthony Sun, Maya Nelson
SoftDev
P00 - Wiki

Program Components

Backend updates

- Sql database population as outlined in database organization
- python function to parse wiki file: *parse_wiki*
 - Input: path/to/wikifile
 - Returns: 2D list of pages referenced, with [page_referenced, (optional) nickname] elements
- python function to update SQLite3 database: *update*
 - Input: directory of wiki files
 - Process:
 - Prunes SQL database of orphaned tables
 - Gets list of page-reference data from *parse_wiki*
 - Creates/updates table for each wikifile in directory
 - Returns: none
 - Issues: how do we recursively search directories?

Rendering

- Use HTML templates and forms and have users fill in specific fields to render the wiki page, with the filled in content. Use python script and Flask to serve this.
- Python function to return contents of index table as HTML: *index_contents*
 - Input: none
 - Returns: contents of index as HTML
- python script to generate webserver: *app*
 - Root ("/") route returns *index_contents()*
 - Variable ("/<page>") route returns *render([page](#))*

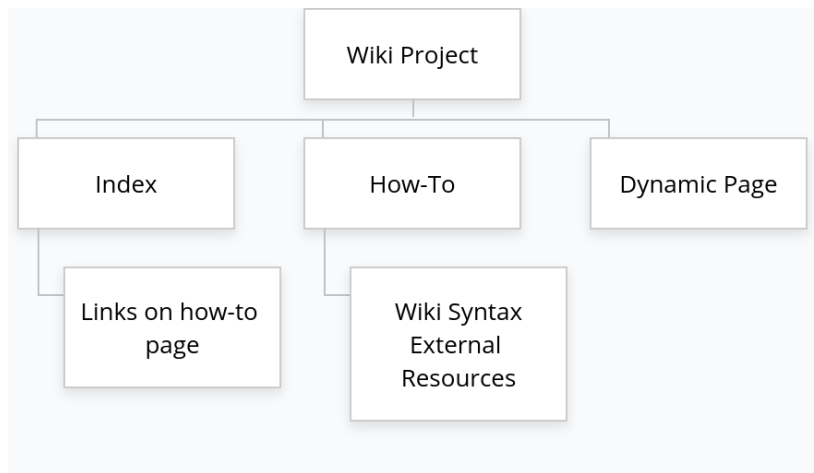
Starter pages

- Ships with how-to docs and sample wiki links

Database Organization

- 1 master table with all wiki pages called "index"
 - TEXT pagename PRIMARY KEY | TEXT path/to/wikifile
- One table for each new page created, titled the name of the file
 - TEXT page_referenced PRIMARY KEY | LIST nicknames
 - nicknames must be unique

Site Map



- Index - list of entries in **index** table
- How-to - both a sample wiki page and a guide to creating wiki pages
 - Wiki syntax
 - External resources
- Dynamic page - updates based on content in wiki

Breakdown of Tasks

Anthony (A); Elizabeth (E); Maya (M)

- E: Write howto/filler text for how-to and its subpages
- E, A: Write parse_wiki
- A, M: Write update
- E, M: Write render
- E, M: Write index_contents
- A: Write app