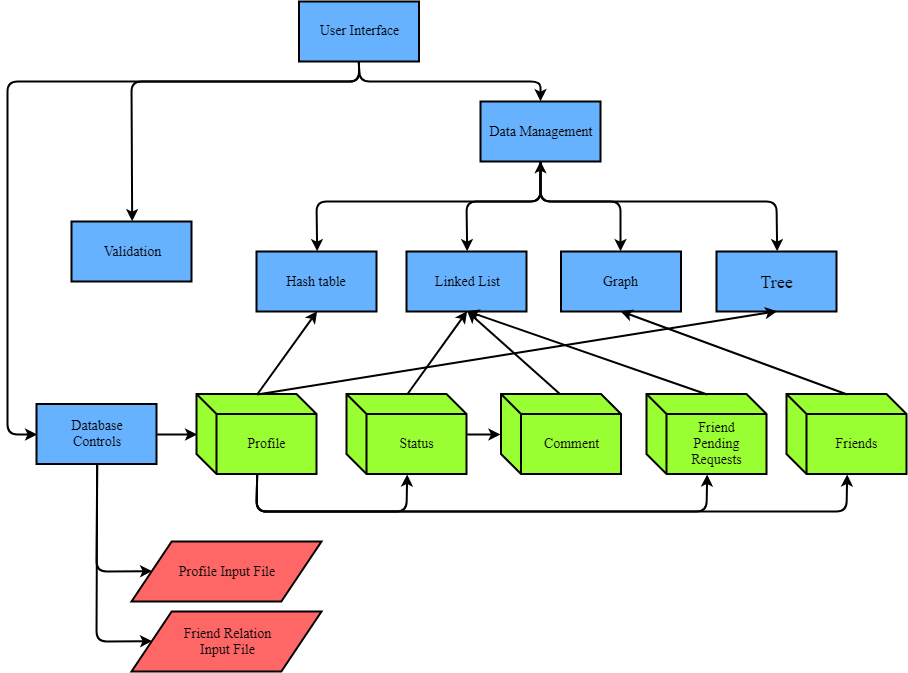
**ARCHITECTURE**



* Each box represents C file.
* The cubes represent data structure.
* The parallelograms represents information input file.

**MODULE DESCRIPTIONS**

|  |  |
| --- | --- |
| **Component** | **Functionality** |
| User Interface | - Get input from users.  - Display input to user |
| Validation | - Validate input from users  - Print error message If input that is validated is not in form.  - Validate input from database |
| Database Controls | - Read information about profile users and friend relations.  - Use this information to create primary data structures.  - Detect errors in the input file format. |
| Data Management | - Manage all of data that keep in different form of structures.  - Put together many functions in all of data structures into one function for easy to use in whole program. |
| Hash table | - Manage data that related to hash table.  - Key string of hash table is E-mail  - If want to add data in same position, it will keep in binary tree structure. |
| Linked List | - Manage data that related to linked list.  - In our program used linked list structure to keep Status, Comment of each status, and Friend Pending Request. |
| Graph | - Manage data that related to graph.  - In our program used graph structure on friend’s relation such as add friend, delete friend, print status of friend.  - Keep friend relation in binary tree structure. |
| Tree | - Manage data that related to binary tree.  - Consider keeping data in left node and right node by compare string, which string that compare is E-mail.  - In our program used tree structure on hash table (when there are same position) and friend relation (graph structure.) |