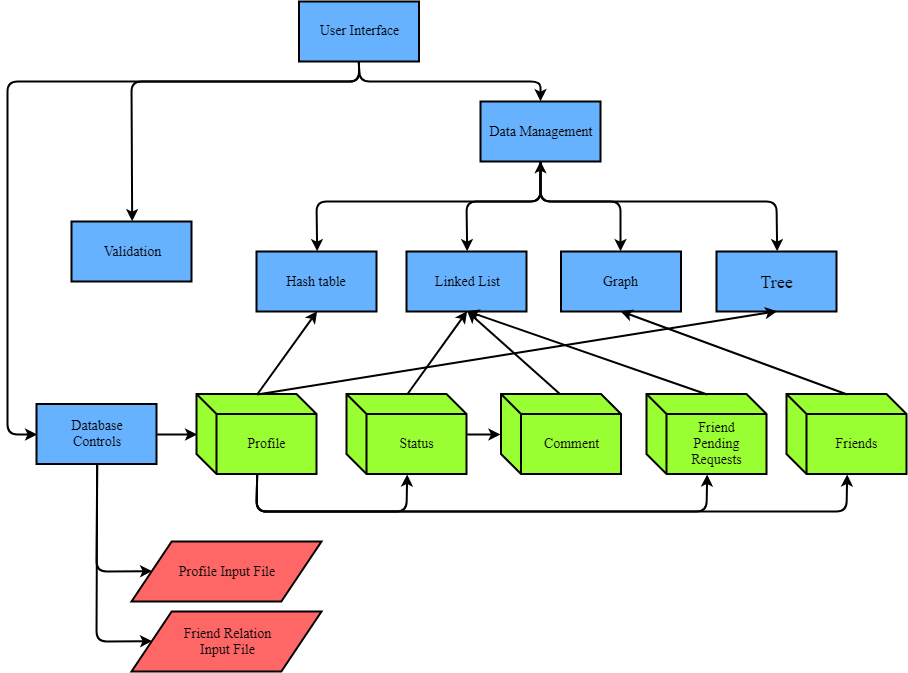
**ARCHITECTURE**



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

**MODULE DESCRIPTIONS**

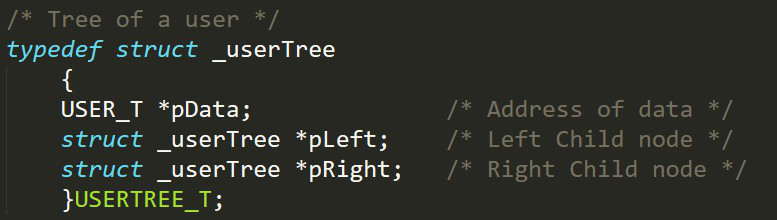
|  |  |
| --- | --- |
| **Component** | **Functionality** |
| User Interface | - Gets input from user.  - Displays input to user. |
| Validation | - Validates input from user.  - Validates input from database.  - Prints error message If input that is validated is not in form. |
| Database Controls | - Reads information about profile users and friend relations.  - Uses this information to create primary data structures.  - Detects errors in the input file format. |
| Data Management | - Manages all of data that keep in different form of structures.  - Puts many functions together in all of data structures into one function for easy to use in whole program. |
| Hash table | - Manages data that related to hash table.  - Key string of hash table is E-mail.  - If user wants to add data in same position, it will keep in binary tree structure. |
| Linked List | - Manages data that related to linked list.  - In our program, uses linked list structure to keep Status, Comment of each status, and Friend Pending Request. |
| Graph | - Manages data that related to graph.  - In our program, uses graph structure on friend’s relation such as adding friend, deleting friend, printing status of friend.  - Keeps friend relation in binary tree structure. |
| Tree | - Manages data that related to binary tree.  - Considers 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.) |

**PRIMARY DATA STRUCTURES**

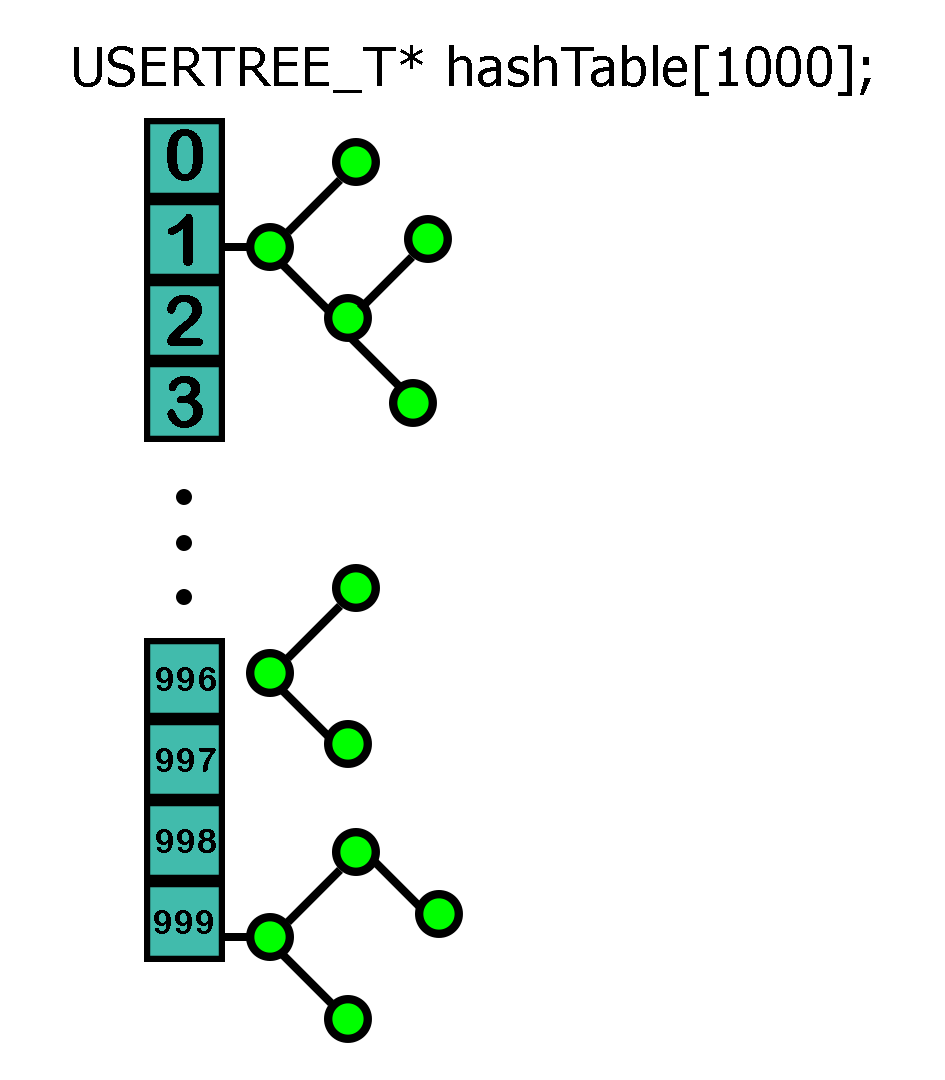
**1. Profile**

- Each structure of profile is kept in Hash table (There are 1,000 arrays in Hash table.). If there is data that want to keep in same position of hash table, that data will be kept by using Trees.

- Structure of tree keeps address of user, left node and right node.

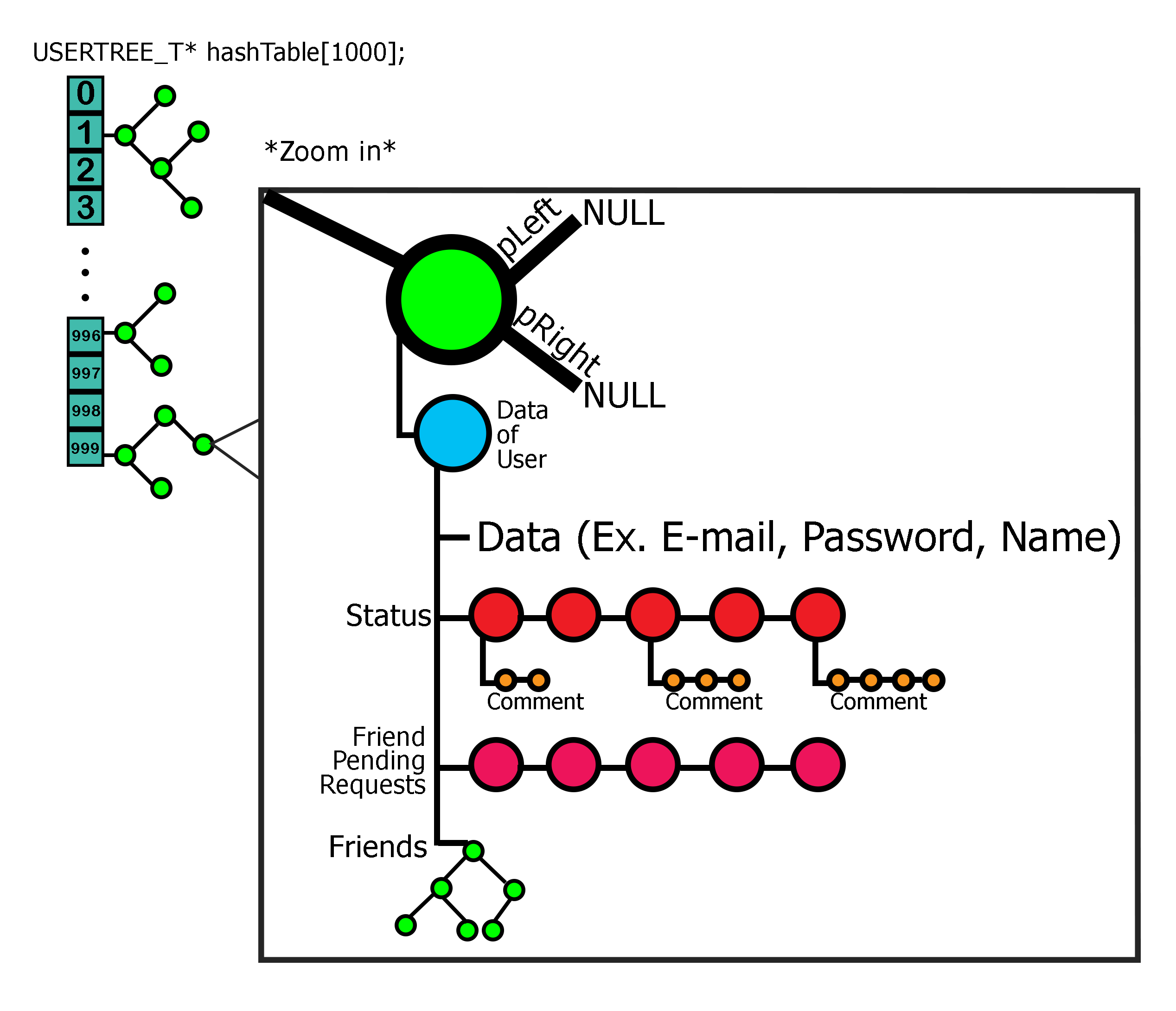


This figure shows about structure of trees.



This figure shows about how to keep structure of profile. Each circle represents each profile.

- Structure of each profile keeps E-mail, password, name, birthday, phone number, gender, friends’ status, friends and friend pending requests.

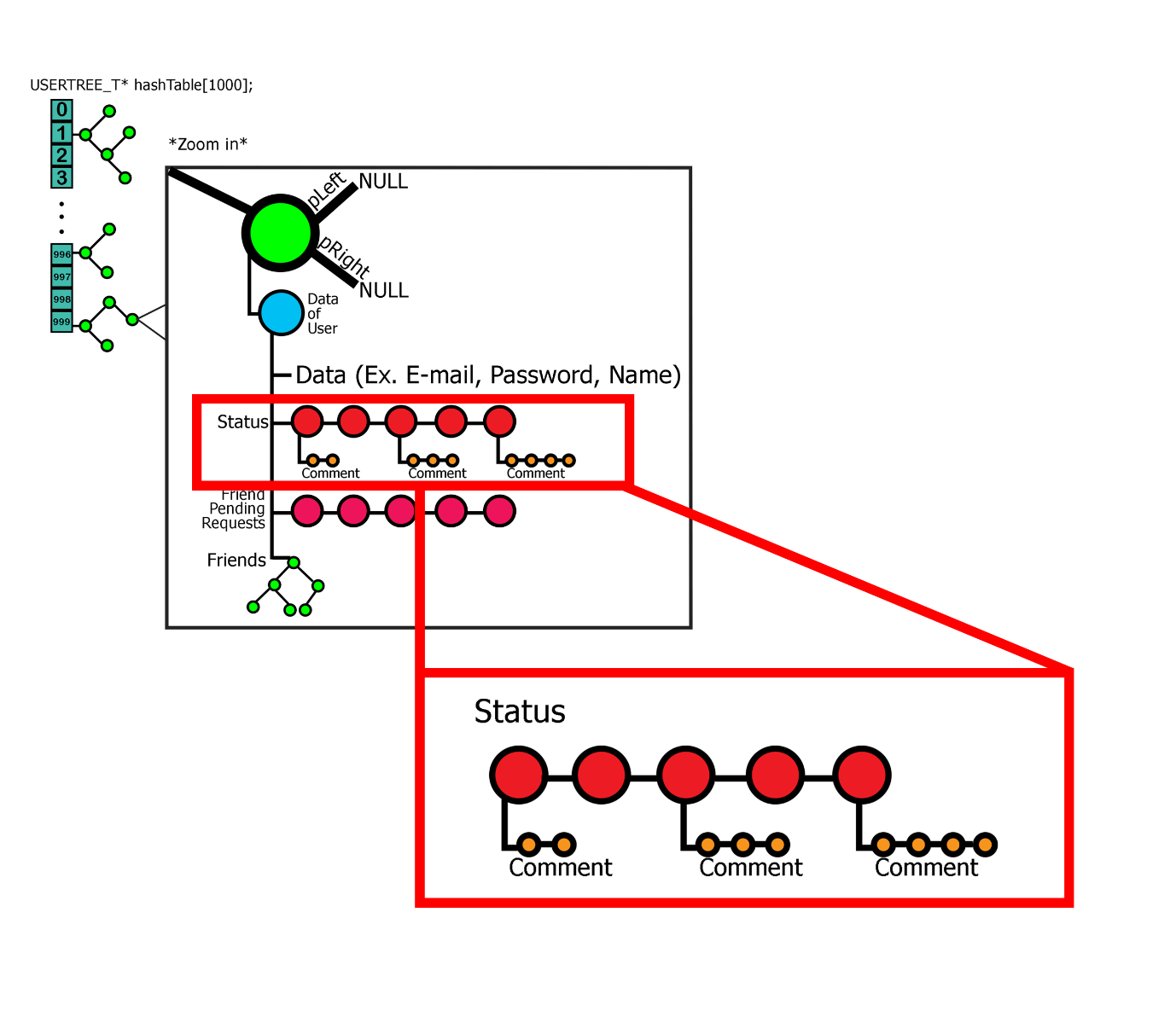


This figure shows about data in structure of profile.

**2. Status**

- Structure of each status keeps text, date, comment, and next node of status.

- Each structure of status is kept by using Linked List.



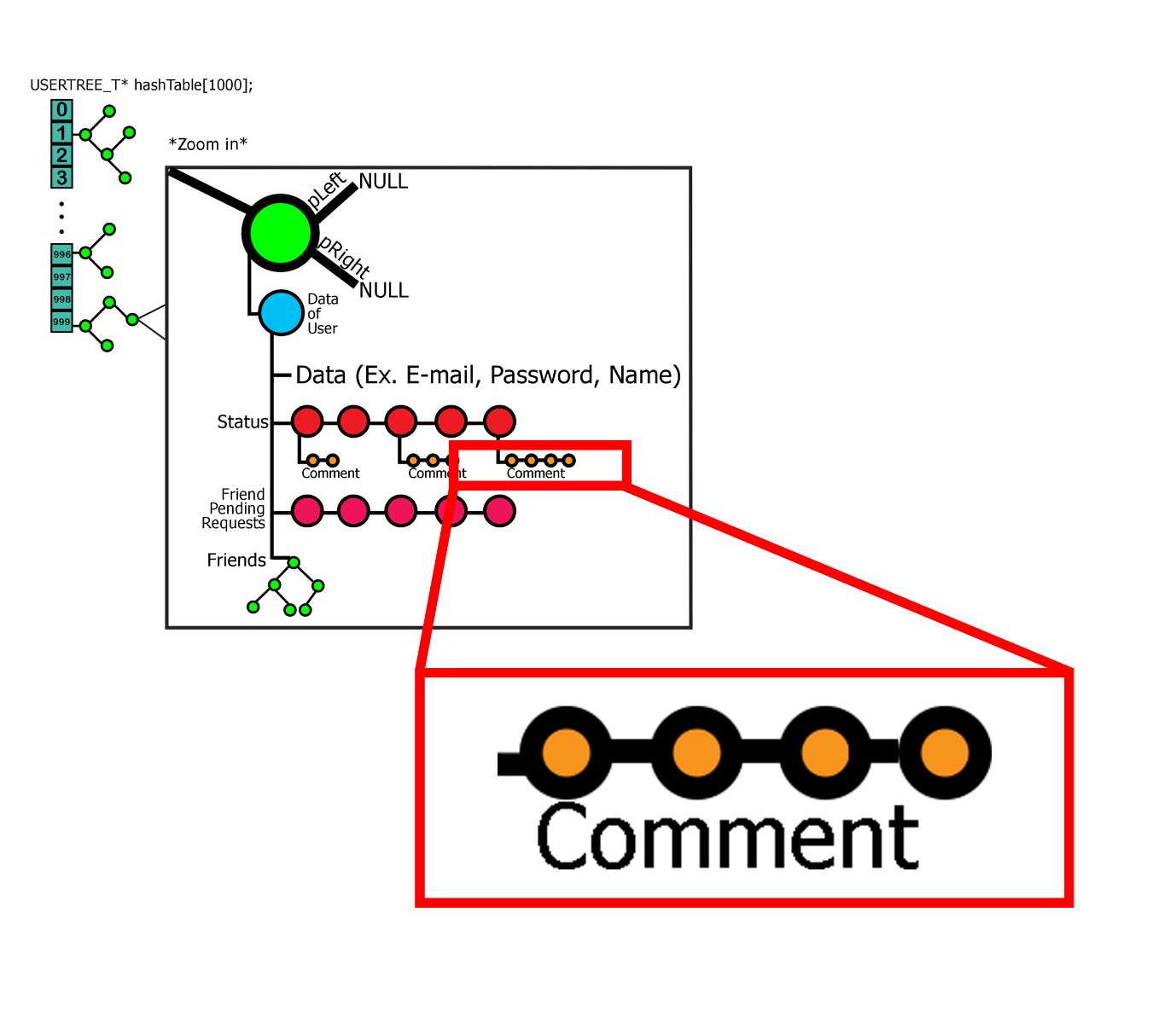
This figure shows about how to keep structure of status.

Each red circle represents structure of status.

**3. Comment**

- Structure of each comment keeps address of owner, text, date, and next node of comment.

- Each structure of comment is kept by using Linked List.



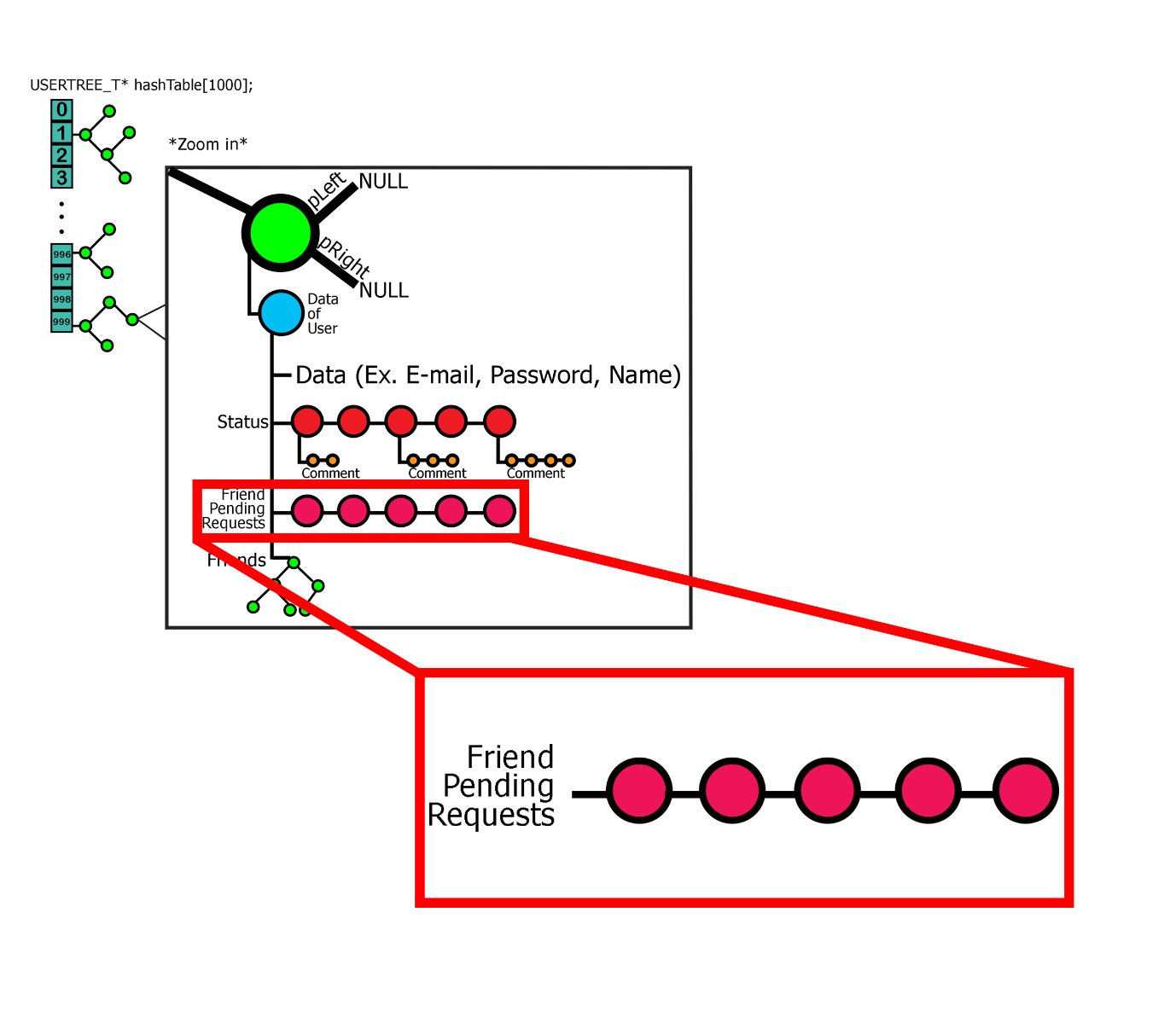
This figure shows about how to keep structure of comment.

Each orange circle represents structure of comment.

**4. Friend Pending Requests**

**-** Structure of friend pending request keeps address of person who wants to be friend and next node of structure.

- Each structure of friend pending request is kept by using Linked List.



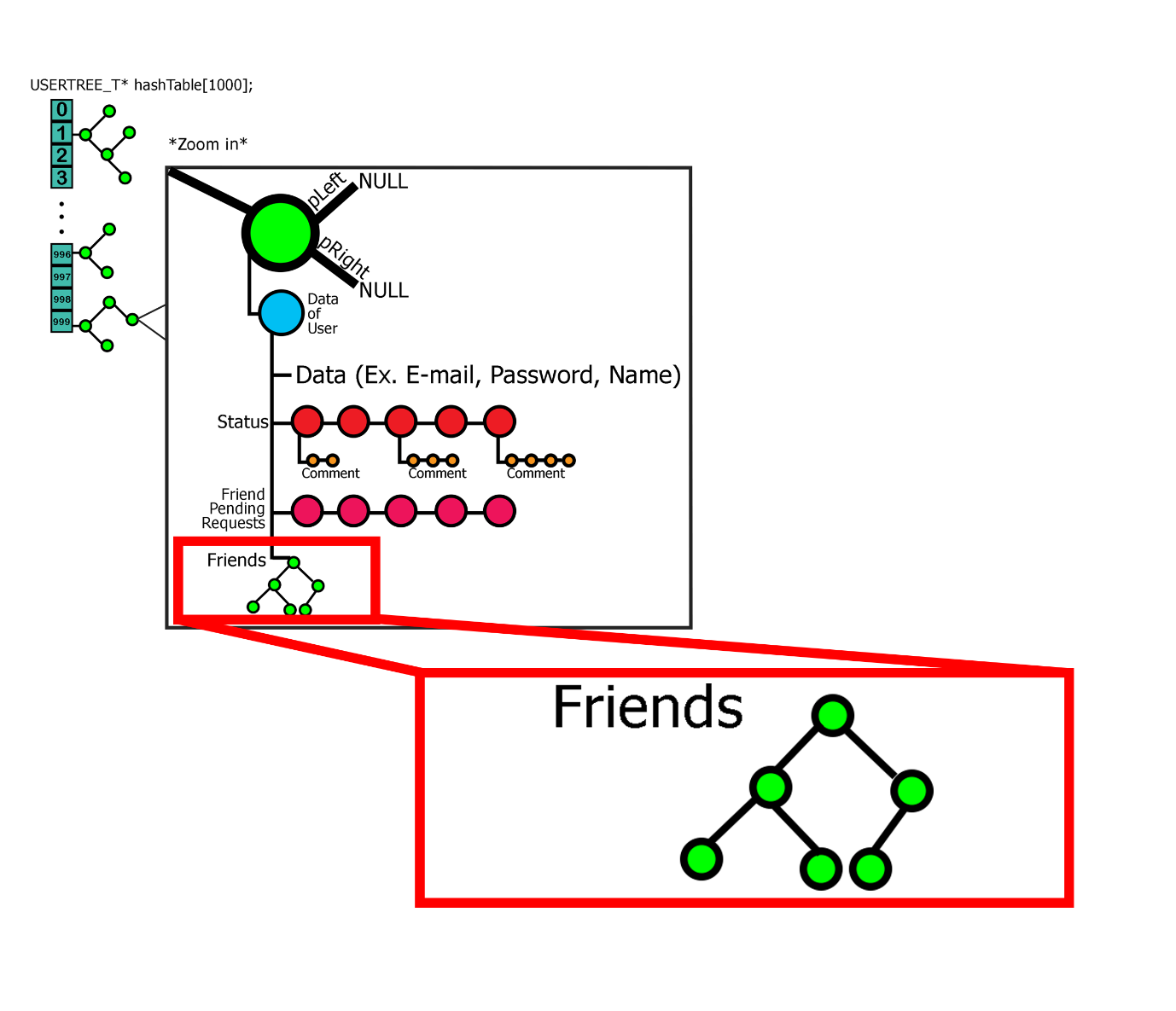
This figure shows about how to keep structure of friend pending request.

Each pink circle represents structure of friend pending request.

**5. Friends**

**-** Structure of friend keeps address of person who is friend.

- Each structure of friend is kept by using Trees.



This figure shows about how to keep structure of friend.

Each green circle represents structure of friend.

**PROJECT DATA FILES**

For the simple facebook, we have 3 data files, One for holding all the profile information, Second one holding status and comment and third one for holding all the friend relations and pending request.

**Profile Input File**

EMAIL [TK@gmail.com](mailto:TK@gmail.com)

PASSWORD GetACPE111

NAME Nathaphop Sundarabhogin

BIRTH 8 10 1998

PHONE 0859095626

GENDER MALE

EMAIL Saab@hotmail.com

PASSWORD DataStructure555

NAME Setthawut Leelawatthanapanit

BIRTH 20 8 1998

PHONE 0900044267

GENDER MALE

...

**EMAIL** – defines an E-mail of each user ( This is unique for each user ). Program will read all next data and keep in this E-mail until program found another E-mail.

**PASSWORD** – defines a password of each account. Used for signs in to Simple Facebook. Each password for last E-mail that program read.

**NAME** – defines Name of user in Simple Facebook. Each name for last E-mail that program read.

**BIRTH** – defines a birth date of user in Simple Facebook. Each Birth date for last E-mail that program read.

**PHONE** – defines a phone number of user in Simple Facebook. Each phone number for last E-mail that program read.

**GENDER** – defines a gender of user in Simple Facebook. Each gender for last E-mail that program read.

**Friend Relations Input File**

FRIEND TK@gmail.com Saab@hotmail.com Tong@gmail.com

FRIEND Saab@hotmail.com TK@gmail.com Cherprang@bnk.in.th

FRIEND Tong@gmail.com Pun@bnk.in.th TK@gmail.com

…

PENDING TK@gmail.com Cherprang@bnk.in.th Pun@bnk.in.th

PENDING Saab@hotmail.com Pun@bnk.in.th Brigth@iaun.com

PENDING Tong@gmail.com Bright@iaun.com

…

**FRIEND** – defines a friend of each account. The first E-mail is E-mail of each user and follows by friends of that account.

**PENDING** – defines a friend pending request of each account. The first E-mail is E-mail of each user and follows by friends that send pending request.

**Status and Comment Input File**

STATUS TK@gmail.com Hello World

DATE 10 1 2018

COMMENT Saab@hotmail.com Hi!

DATE 12 2 2018

STATUS TK@gmail.com Get A CPE111

DATE 20 2 2018

COMMENT Tong@gmail.com Really?

DATE 21 2 2018

COMMENT Saab@hotmail.com Cheer!

DATE 22 2 2018

…

**STATUS** – defines a user who update status and text. The first E-mail is E-mail of user that update status and follows by text that user update.

**PENDING** – defines a friend pending request of each account. The first E-mail is E-mail of each user and follows by friends that send pending request

**DATE** – defines a date when update status/ comment. Depends on what is before. If status is before date, this date will keep in that status. But if comment is before date, this date will keep in comment. Keep only 3 numbers (Date Month Year)