Trello_Project_Management_System_

1.0

Generated by Doxygen 1.13.2

1 Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

2 File Index

2.1 File List

Here is a list of all files with brief descriptions:

3 Data Structure Documentation

3.1 CARD Struct Reference

Represents a Card in Trello_Project_Management_System.

```
#include <cards.h>
```

Data Fields

- char name [MAX_NAME_LENGTH]
- char description [MAX_DESCRIPTION_LENGTH]
- struct CARD ∗ next_card
- struct CARD ∗ prev_card

3.1.1 Detailed Description

Represents a Card in Trello_Project_Management_System.

3.1.2 Field Documentation

description

```
char description[MAX_DESCRIPTION_LENGTH]
```

CARD description

name

```
char name[MAX_NAME_LENGTH]
```

CARD name

next_card

```
struct CARD∗ next_card
```

A pointer to the next card

prev_card

```
struct CARD∗ prev_card
```

A pointer to the previous card

The documentation for this struct was generated from the following file:

- cards.h

3.2 LIST Struct Reference

Represents a List in Trello_Project_Management_System.

```
#include <lists.h>
```

Data Fields

- char name [MAX_NAME_LENGTH]
- CARD ∗ CARD
- struct LIST ∗ next_List
- struct LIST ∗ prev_List

3.2.1 Detailed Description

Represents a List in Trello_Project_Management_System.

3.2.2 Field Documentation

CARD

```
CARD* CARD
```

A pointer to the first card of the list

name

```
char name[MAX_NAME_LENGTH]
```

LIST name

next_List

```
struct LIST* next_List
```

A pointer to the next list

prev_List

```
struct LIST* prev_List
```

A pointer to the previous list

The documentation for this struct was generated from the following file:

- lists.h

3.3 USER Struct Reference

Represents a identity of a user.

```
#include <user_entry.h>
```

Data Fields

- char name [MAX_LENGTH]
- char email_id [MAX_LENGTH]

3.3.1 Detailed Description

Represents a identity of a user.

3.3.2 Field Documentation

email_id

```
char email_id[MAX_LENGTH]
```

USER email id

name

```
char name[MAX_LENGTH]
```

USER name

The documentation for this struct was generated from the following file:

- user_entry.h

4 File Documentation

4.1 cards.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <stdbool.h>
#include "cards.h"
```

Functions

- CARD ∗ create_Card (char name[MAX_NAME_LENGTH], char description[MAX_DESCRIPTION_LENGTH])

  Creates a new card in the system.
- void add_a_Card (CARD ∗head)

  Insert a new card after the existing one.
- CARD ∗ delete_a_Card (CARD ∗head)

  Delete any of the existing card from the system.
- CARD ∗ delete_First_Card (CARD ∗head)

  Delete the very first card from the system.
- CARD ∗ delete_Intermediate_or_Last_Card (CARD ∗head)

  Delete any of the existing card from the system without the first one.
- bool checkList ()

  Check whether the task of this card is done or not.

4.1.1 Function Documentation

add_a_Card()

```
void add_a_Card (
            CARD * head)
```

Insert a new card after the existing one.

Parameters

head	The pointer to the very first card

checkList()

```
bool checkList ()
```

Check whether the task of this card is done or not.

Returns

DONE if the task has been completed, NOT_DONE if that is incomplete

create_Card()

```
CARD * create_Card (
            char name[MAX_NAME_LENGTH],
            char description[MAX_DESCRIPTION_LENGTH])
```

Creates a new card in the system.

Parameters

name	The unique name of the card
description	The about information of the new card

Returns

A pointer to the new card

delete_a_Card()

```
CARD * delete_a_Card (
            CARD * head)
```

Delete any of the existing card from the system.

Parameters

head	The pointer to the very first card

Returns

A pointer to the new first card, if the very first card removed

delete_First_Card()

```
CARD * delete_First_Card (
            CARD * head)
```

Delete the very first card from the system.

Parameters

head	The pointer to the existing first card

Returns

A pointer to the new first card

delete_Intermediate_or_Last_Card()

```
CARD * delete_Intermediate_or_Last_Card (
            CARD * head)
```

Delete any of the existing card from the system without the first one.

Parameters

head	The pointer to the very first card

Returns

A pointer to the first card

4.2 cards.h File Reference

Header for managing the cards in the Tello_Project_Management_System.

```
#include <stdbool.h>
```

Data Structures

- struct CARD

  Represents a Card in Trello_Project_Management_System.

Macros

- #define MAX_DESCRIPTION_LENGTH 250

  Maximum number of characters allowed for writing description.
- #define MAX_NAME_LENGTH 30

  Maximum number of characters allowed for writing name.
- #define DONE 1

  To make the code readable, function will return DONE means 1.
- #define NOT_DONE 0

  To make the code readable, function will return NOT_DONE means 0.

Typedefs

- typedef struct CARD CARD

  Represents a Card in Trello_Project_Management_System.

Functions

- CARD ∗ create_Card (char name[MAX_NAME_LENGTH], char description[MAX_DESCRIPTION_LENGTH])

  Creates a new card in the system.
- void add_a_Card (CARD ∗head)

  Insert a new card after the existing one.
- CARD ∗ delete_a_Card (CARD ∗head)

  Delete any of the existing card from the system.
- CARD ∗ delete_First_Card (CARD ∗head)

  Delete the very first card from the system.
- CARD ∗ delete_Intermediate_or_Last_Card (CARD ∗head)

  Delete any of the existing card from the system without the first one.
- bool checkList ()

  Check whether the task of this card is done or not.

4.2.1 Detailed Description

Header for managing the cards in the Tello_Project_Management_System.

Author

Data_Structures_Project_Group_01

4.2.2 Macro Definition Documentation

DONE

```
#define DONE 1
```

To make the code readable, function will return DONE means 1.

MAX_DESCRIPTION_LENGTH

```
#define MAX_DESCRIPTION_LENGTH 250
```

Maximum number of characters allowed for writing description.

MAX_NAME_LENGTH

```
#define MAX_NAME_LENGTH 30
```

Maximum number of characters allowed for writing name.

NOT_DONE

```
#define NOT_DONE 0
```

To make the code readable, function will return NOT_DONE means 0.

4.2.3 Typedef Documentation

CARD

```
typedef struct CARD CARD
```

Represents a Card in Trello_Project_Management_System.

4.2.4 Function Documentation

add_a_Card()

```
void add_a_Card (
            CARD * head)
```

Insert a new card after the existing one.

Parameters

head	The pointer to the very first card

checkList()

```
bool checkList ()
```

Check whether the task of this card is done or not.

Returns

DONE if the task has been completed, NOT_DONE if that is incomplete

create_Card()

```
CARD * create_Card (
            char name[MAX_NAME_LENGTH],
            char description[MAX_DESCRIPTION_LENGTH])
```

Creates a new card in the system.

Parameters

name	The unique name of the card
description	The about information of the new card

Returns

A pointer to the new card

delete_a_Card()

```
CARD * delete_a_Card (
            CARD * head)
```

Delete any of the existing card from the system.

Parameters

head	The pointer to the very first card

Returns

A pointer to the new first card, if the very first card removed

delete_First_Card()

```
CARD * delete_First_Card (
            CARD * head)
```

Delete the very first card from the system.

Parameters

head	The pointer to the existing first card

Returns

A pointer to the new first card

delete_Intermediate_or_Last_Card()

```
CARD * delete_Intermediate_or_Last_Card (
            CARD * head)
```

Delete any of the existing card from the system without the first one.

Parameters

head	The pointer to the very first card

Returns

A pointer to the first card

4.3 cards.h

Go to the documentation of this file.

```
#ifndef CARDS_H_INCLUDED
#define CARDS_H_INCLUDED



#define MAX_DESCRIPTION_LENGTH 250

#define MAX_NAME_LENGTH 30

#define DONE 1

#define NOT_DONE 0
#include<stdbool.h>
typedef struct CARD
{
    char name[MAX_NAME_LENGTH];
    char description[MAX_DESCRIPTION_LENGTH];
    struct CARD* next_card;
    struct CARD* prev_card;
}CARD;

CARD* create_Card(char name[MAX_NAME_LENGTH],char description[MAX_DESCRIPTION_LENGTH]);

void add_a_Card(CARD* head);

CARD* delete_a_Card(CARD* head);

CARD* delete_First_Card(CARD* head);

CARD* delete_Intermediate_or_Last_Card(CARD* head);

bool checkList();
#endif
```

4.4 lists.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <stdbool.h>
#include "lists.h"
#include "cards.h"
```

Functions

- LIST ∗ create_List (char name[MAX_NAME_LENGTH])

  Creates a new list in the system.
- void add_a_List (LIST ∗head)

  Insert a new list after the existing one.
- LIST ∗ delete_a_List (LIST ∗head)

  Delete any of the existing list from the system.
- LIST ∗ delete_First_List (LIST ∗head)

  Delete the very first list from the system.
- LIST ∗ delete_Intermediate_or_Last_List (LIST ∗head)

  Delete any of the existing list from the system without the first one.

4.4.1 Function Documentation

add_a_List()

```
void add_a_List (
          LIST ∗ head)
```

Insert a new list after the existing one.

Parameters

head	The pointer to the very first card

create_List()

```
LIST ∗ create_List (
          char name[MAX_NAME_LENGTH])
```

Creates a new list in the system.

Parameters

name	The unique name of the list

Returns

A pointer to the new list

delete_a_List()

```
LIST * delete_a_List (
             LIST * head)
```

Delete any of the existing list from the system.

Parameters

head	The pointer to the very first list

Returns

A pointer to the new first list, if the very first list removed

delete_First_List()

```
LIST * delete_First_List (
            LIST * head)
```

Delete the very first list from the system.

Parameters

head	The pointer to the existing first list

Returns

A pointer to the new first list

delete_Intermediate_or_Last_List()

```
LIST * delete_Intermediate_or_Last_List (
            LIST * head)
```

Delete any of the existing list from the system without the first one.

Parameters

head	The pointer to the very first list

Returns

A pointer to the first list

4.5 lists.h File Reference

Header for managing the lists in the Tello_Project_Management_System.

```
#include "cards.h"
```

Data Structures

- struct LIST

  Represents a List in Trello_Project_Management_System.

Macros

- #define MAX_NAME_LENGTH 30

  Maximum number of characters allowed for writing name.

Typedefs

- typedef struct LIST LIST

  Represents a List in Trello_Project_Management_System.

Functions

- LIST ∗ create_List (char name[MAX_NAME_LENGTH])

  Creates a new list in the system.
- void add_a_List (LIST ∗head)

  Insert a new list after the existing one.
- LIST ∗ delete_a_List (LIST ∗head)

  Delete any of the existing list from the system.
- LIST ∗ delete_First_List (LIST ∗head)

  Delete the very first list from the system.
- LIST ∗ delete_Intermediate_or_Last_List (LIST ∗head)

  Delete any of the existing list from the system without the first one.

4.5.1 Detailed Description

Header for managing the lists in the Tello_Project_Management_System.

Author

Data_Structures_Project_Group_01

4.5.2 Macro Definition Documentation

MAX_NAME_LENGTH

```
#define MAX_NAME_LENGTH 30
```

Maximum number of characters allowed for writing name.

4.5.3 Typedef Documentation

LIST

```
typedef struct LIST LIST
```

Represents a List in Trello_Project_Management_System.

4.5.4 Function Documentation

add_a_List()

```
void add_a_List (
             LIST * head)
```

Insert a new list after the existing one.

Parameters

head	The pointer to the very first card

create_List()

```
LIST * create_List (
             char name[MAX_NAME_LENGTH])
```

Creates a new list in the system.

Parameters

name	The unique name of the list

Returns

A pointer to the new list

delete_a_List()

```
LIST * delete_a_List (
             LIST * head)
```

Delete any of the existing list from the system.

Parameters

head	The pointer to the very first list

Returns

A pointer to the new first list, if the very first list removed

delete_First_List()

```
LIST * delete_First_List (
             LIST * head)
```

Delete the very first list from the system.

Parameters

head	The pointer to the existing first list

Returns

A pointer to the new first list

delete_Intermediate_or_Last_List()

```
LIST * delete_Intermediate_or_Last_List (
            LIST * head)
```

Delete any of the existing list from the system without the first one.

Parameters

head	The pointer to the very first list

Returns

A pointer to the first list

4.6 lists.h

Go to the documentation of this file.

```
#ifndef LISTS_H_INCLUDED
#define LISTS_H_INCLUDED


#include"cards.h"

#define MAX_NAME_LENGTH 30

typedef struct LIST
{
    char name[MAX_NAME_LENGTH];
    CARD* CARD;
    struct LIST* next_List;
    struct LIST* prev_List;
}LIST;

LIST* create_List(char name[MAX_NAME_LENGTH]);

void add_a_List(LIST* head);

LIST* delete_a_List(LIST* head);

LIST* delete_First_List(LIST* head);

LIST* delete_Intermediate_or_Last_List(LIST* head);
#endif
```

4.7 search.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "lists.h"
#include "cards.h"
#include "search.h"
```

Functions

- void search_Card (CARD ∗head, char desired_Card[MAX_NAME_LENGTH])

  Search the desired card from the system.
- void search_List (LIST ∗head, char desired_List[MAX_NAME_LENGTH])

  Search the desired list from the system.

4.7.1 Function Documentation

search_Card()

```
void search_Card (
            CARD * head,
            char desired_Card[MAX_NAME_LENGTH])
```

Search the desired card from the system.

Parameters

head	The pointer to the very first card
desired_List	The name of the card to search

search_List()

```
void search_List (
            LIST * head,
            char desired_List[MAX_NAME_LENGTH])
```

Search the desired list from the system.

Parameters

head	The pointer to the very first list
desired_List	The name of the list to search

4.8 search.h File Reference

Header for searching the cards or lists in the Tello_Project_Management_System.

```
#include "lists.h"
#include "cards.h"
```

Functions

- void search_List (LIST ∗head, char desired_List[MAX_NAME_LENGTH])

  Search the desired list from the system.
- void search_Card (CARD ∗head, char desired_Card[MAX_NAME_LENGTH])

  Search the desired card from the system.

4.8.1 Detailed Description

Header for searching the cards or lists in the Tello_Project_Management_System.

Author

Data_Structures_Project_Group_01

4.8.2 Function Documentation

search_Card()

```
void search_Card (
            CARD * head,
            char desired_Card[MAX_NAME_LENGTH])
```

Search the desired card from the system.

Parameters

head	The pointer to the very first card
desired_List	The name of the card to search

search_List()

```
void search_List (
            LIST * head,
            char desired_List[MAX_NAME_LENGTH])
```

Search the desired list from the system.

Parameters

head	The pointer to the very first list
desired_List	The name of the list to search

4.9 search.h

Go to the documentation of this file.

```
#ifndef SEARCH_H_INCLUDED
#define SEARCH_H_INCLUDED


#include"lists.h"
#include"cards.h"

void search_List(LIST* head,char desired_List[MAX_NAME_LENGTH]);

void search_Card(CARD* head, char desired_Card[MAX_NAME_LENGTH]);
#endif
```

4.10 user_entry.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "user_entry.h"
```

Functions

- void registration ()

  Store the information of the user for further Log In.
- int log_In ()

  Compare the stored information of the user from file to provide access.

4.10.1 Function Documentation

log_In()

```
int log_In ()
```

Compare the stored information of the user from file to provide access.

registration()

```
void registration ()
```

Store the information of the user for further Log In.

4.11 user_entry.h File Reference

Header for managing user entry.

Data Structures

- struct USER

  Represents a identity of a user.

Macros

- #define MAX_LENGTH 16

  Maximum number of characters allowed for name and email-id.

Typedefs

- typedef struct USER USER

  Represents a identity of a user.

Functions

- void registration ()

  Store the information of the user for further Log In.
- int log_In ()

  Compare the stored information of the user from file to provide access.

4.11.1 Detailed Description

Header for managing user entry.

Author

Data_Structures_Project_Group_01

4.11.2 Macro Definition Documentation

MAX_LENGTH

```
#define MAX_LENGTH 16
```

Maximum number of characters allowed for name and email-id.

4.11.3 Typedef Documentation

USER

```
typedef struct USER USER
```

Represents a identity of a user.

4.11.4 Function Documentation

log_In()

```
int log_In ()
```

Compare the stored information of the user from file to provide access.

registration()

```
void registration ()
```

Store the information of the user for further Log In.

4.12 user_entry.h

Go to the documentation of this file.

```
#ifndef USER_ENTRY_H_INCLUDED
#define USER_ENTRY_H_INCLUDED


#define MAX_LENGTH 16

typedef struct USER
{
    char name[MAX_LENGTH];
    char email_id[MAX_LENGTH];
}USER;

void registration();

int log_In();
#endif
```

Index