

Документация на использование программы сервера

Создано системой Doxygen 1.9.4



1 Алфавитный указатель классов	1
1.1 Классы	1
2 Список файлов	3
2.1 Файлы	3
3 Классы	5
3.1 Класс DataReader	5
3.1.1 Подробное описание	5
3.1.2 Методы	5
3.1.2.1 getClient()	5
3.2 Структура DataReader_fix	6
3.3 Класс Errors	6
3.3.1 Подробное описание	7
3.3.2 Методы	7
3.3.2.1 error_recording()	7
3.4 Структура Errors_fix	8
3.5 Структура Result	8
3.6 Структура Server_fix	9
3.7 Класс ServerClientInterface	9
3.7.1 Подробное описание	10
3.7.2 Методы	10
3.7.2.1 sum_vec()	10
3.8 Класс User	10
3.8.1 Подробное описание	11
3.8.2 Методы	11
3.8.2.1 CheckLogin()	11
3.8.2.2 CheckPassword()	12
3.9 Структура User_fix	12
4 Файлы	13
4.1 Файл DataReader.cpp	13
4.2 Файл DataReader.h	13
4.2.1 Подробное описание	14
4.3 DataReader.h	15
4.4 Файл Errors.cpp	15
4.5 Файл Errors.h	16
4.5.1 Подробное описание	17
4.6 Errors.h	17
4.7 Файл main.cpp	17
4.7.1 Подробное описание	18
4.8 Файл ServerClientInterface.cpp	18
4.9 Файл ServerClientInterface.h	18
4.9.1 Подробное описание	19

4.10 ServerClientInterface.h . . . . .	20
4.11 Файл User.cpp . . . . .	21
4.12 Файл User.h . . . . .	21
4.12.1 Подробное описание . . . . .	22
4.13 User.h . . . . .	23
Предметный указатель . . . . .	25

# Глава 1

## Алфавитный указатель классов

### 1.1 Классы

Классы с их кратким описанием.

<a href="#">DataReader</a>	
Класс для получения БД клиентов	5
<a href="#">DataReader_fix</a>	6
<a href="#">Errors</a>	
Класс обработки ошибок	6
<a href="#">Errors_fix</a>	8
<a href="#">Result</a>	8
<a href="#">Server_fix</a>	9
<a href="#">ServerClientInterface</a>	
Класс для взаимодействия сервера с клиентами	9
<a href="#">User</a>	
Класс, представляющий информацию о подключенном пользователе	10
<a href="#">User_fix</a>	12



## Глава 2

# Список файлов

### 2.1 Файлы

Полный список документированных файлов.

<a href="#">DataReader.cpp</a>	13
<a href="#">DataReader.h</a>	
Заголовочный файл для модуля <a href="#">DataReader</a>	13
<a href="#">Errors.cpp</a>	15
<a href="#">Errors.h</a>	
Заголовочный файл для модуля <a href="#">Errors</a>	16
<a href="#">main.cpp</a>	
Функция для предоставления пользователю справки по использованию программы	17
<a href="#">ServerClientInterface.cpp</a>	18
<a href="#">ServerClientInterface.h</a>	
Заголовочный файл для модуля <a href="#">ServerClientInterface</a>	18
<a href="#">User.cpp</a>	21
<a href="#">User.h</a>	
Заголовочный файл для модуля <a href="#">User</a>	21





## Глава 3

# Классы

### 3.1 Класс DataReader

Класс для получения БД клиентов

```
#include <DataReader.h>
```

Открытые члены

- `string get_FileReader ()`  
Геттер для атрибута FileReader.
- `void set_FileReader (string file)`  
Сеттер для атрибута FileReader.
- `pair< vector< string >, vector< string > > getClient ()`  
Функция для получения БД клиентов

#### 3.1.1 Подробное описание

Класс для получения БД клиентов

Аргументы

FileReader	путь к файлу с БД клиентов
Err	объект класса <a href="#">Errors</a> обработки ошибок

#### 3.1.2 Методы

##### 3.1.2.1 getClient()

```
pair< vector< string >, vector< string > > DataReader::getClient ( )
```

Функция для получения БД клиентов

Возвращает

Возвращает два вектора: логины и пароли клиентов

Исключения

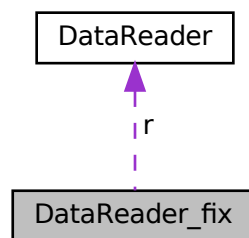
<a href="#">Errors</a>	при возникновении ошибки
------------------------	--------------------------

Объявления и описания членов классов находятся в файлах:

- [DataReader.h](#)
- [DataReader.cpp](#)

## 3.2 Структура DataReader\_fix

Граф связей класса DataReader\_fix:



Открытые атрибуты

- [DataReader](#) \* r

Объявления и описания членов структуры находятся в файле:

- [unit.cpp](#)

## 3.3 Класс Errors

Класс обработки ошибок

```
#include <Errors.h>
```

## Открытые члены

- `string get_File_Log ()`  
Геттер для атрибута `File_Log`.
- `void set_File_Log (string file)`  
Сеттер для атрибута `File_Log`.
- `void error\_recording (string flag, string info)`  
Функция сохранения ошибок в журнал

### 3.3.1 Подробное описание

#### Класс обработки ошибок

##### Аргументы

<code>File_Log</code>	путь к файлу с журналом ошибок
-----------------------	--------------------------------

### 3.3.2 Методы

#### 3.3.2.1 `error_recording()`

```
void Errors::error_recording (  
    string flag,  
    string info )
```

#### Функция сохранения ошибок в журнал

##### Аргументы

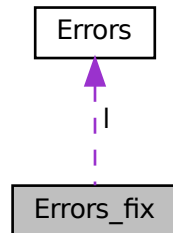
<code>flag</code>	Хранит в себе тип ошибки
<code>info</code>	Хранит в себе подробную информацию об ошибке

Объявления и описания членов классов находятся в файлах:

- [Errors.h](#)
- [Errors.cpp](#)

### 3.4 Структура Errors\_fix

Граф связей класса Errors\_fix:



Открытые атрибуты

- [Errors](#) \* 1

Объявления и описания членов структуры находятся в файле:

- [unit.cpp](#)

### 3.5 Структура Result

Открытые атрибуты

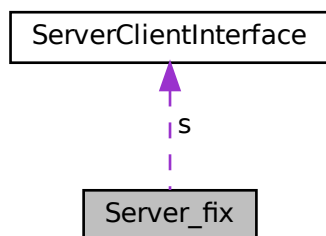
- `std::string hash`
- `std::string salt`
- `std::string login`

Объявления и описания членов структуры находятся в файле:

- [ServerClientInterface.cpp](#)

## 3.6 Структура Server\_fix

Граф связей класса Server\_fix:



Открытые атрибуты

- `ServerClientInterface * s`

Объявления и описания членов структуры находятся в файле:

- `unit.cpp`

## 3.7 Класс ServerClientInterface

Класс для взаимодействия сервера с клиентами

```
#include <ServerClientInterface.h>
```

Открытые члены

- `int interaction (string database, string logFile)`
- `uint64_t sum_vec ()`  
Функция вычисления суммы векторов
- `string get_address ()`  
Геттер для атрибута address.
- `void set_address (string address1)`  
Сеттер для атрибута address.
- `int get_port ()`  
Геттер для атрибута port.
- `void set_port (int port1)`  
Сеттер для атрибута port.
- `vector< uint64_t > get_vec ()`  
Сеттер для атрибута vec.

- `void set_vec (vector< uint64_t > v)`  
Геттер для атрибута `vec`.
- `string get_salt ()`  
Геттер для атрибута `salt`.
- `void set_salt ()`  
Сеттер для атрибута `salt`.
- `pair< vector< string >, vector< string > > get_DB_clients ()`  
Геттер для атрибута `DB_clients`.
- `void set_DB_clients (vector< string > login, vector< string > password)`  
Сеттер для атрибута `DB_clients`.

### 3.7.1 Подробное описание

Класс для взаимодействия сервера с клиентами

Аргументы

<code>address</code>	Адрес сервера
<code>port</code>	Порт сервера
<code>vec</code>	Вектор для вычисления расчетов
<code>DB_clients</code>	БД клиентов
<code>Err</code>	объект класса <a href="#">Errors</a> обработки ошибок

### 3.7.2 Методы

#### 3.7.2.1 `sum_vec()`

```
uint64_t ServerClientInterface::sum_vec ( )
```

Функция вычисления суммы векторов

Возвращает

Возвращает сумму векторов атрибута `vec`

Объявления и описания членов классов находятся в файлах:

- [ServerClientInterface.h](#)
- [ServerClientInterface.cpp](#)

## 3.8 Класс User

Класс, представляющий информацию о подключенном пользователе

```
#include <User.h>
```

## Открытые члены

- bool **CheckLogin** (vector< string > Db\_ID)  
Функция для проверки ID подключенного клиента
- bool **CheckPassword** (vector< string > Db\_password, vector< string > Db\_ID, string SALT, string SendHash)  
Функция для проверки пароля подключенного клиента
- string get\_ID ()  
Геттер для атрибута ID.
- void set\_ID (string ID1)  
Сеттер для атрибута ID.
- string get\_hash ()  
Геттер для атрибута hash.
- void set\_hash (string hash1)  
Сеттер для атрибута hash.

### 3.8.1 Подробное описание

Класс, представляющий информацию о подключенном пользователе

Аргументы

ID	ID подключенного клиента
hash	хэш-код подключенного клиента

### 3.8.2 Методы

#### 3.8.2.1 CheckLogin()

```
bool User::CheckLogin (  
    vector< string > Db_ID )
```

Функция для проверки ID подключенного клиента

Возвращает

Возвращает true - если в БД есть ID подключенного клиента, иначе false

### 3.8.2.2 CheckPassword()

```
bool User::CheckPassword (
    vector< string > Db_password,
    vector< string > Db_ID,
    string SALT,
    string SendHash )
```

Функция для проверки пароля подключенного клиента

Возвращает

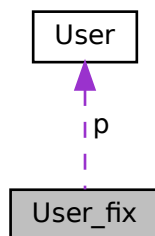
Возвращает true - если пароль подключенного клиента совпадает с паролем из БД, иначе false

Объявления и описания членов классов находятся в файлах:

- [User.h](#)
- [User.cpp](#)

## 3.9 Структура User\_fix

Граф связей класса User\_fix:



Открытые атрибуты

- [User](#) \* p

Объявления и описания членов структуры находятся в файле:

- [unit.cpp](#)



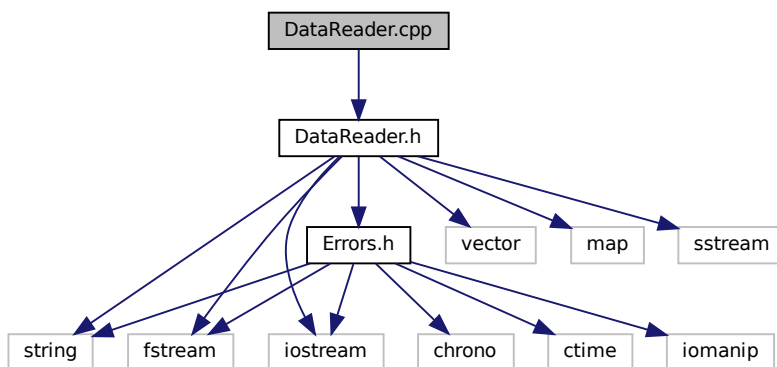
## Глава 4

# Файлы

### 4.1 Файл DataReader.cpp

```
#include "DataReader.h"
```

Граф включаемых заголовочных файлов для DataReader.cpp:



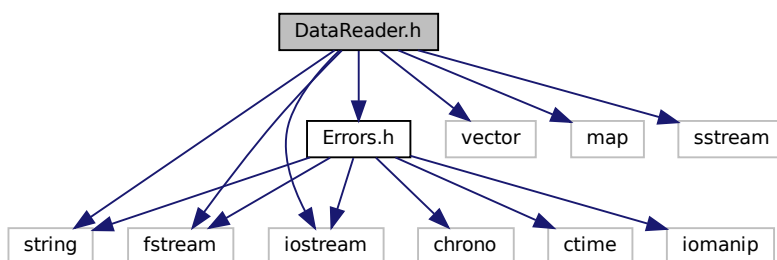
### 4.2 Файл DataReader.h

Заголовочный файл для модуля [DataReader](#).

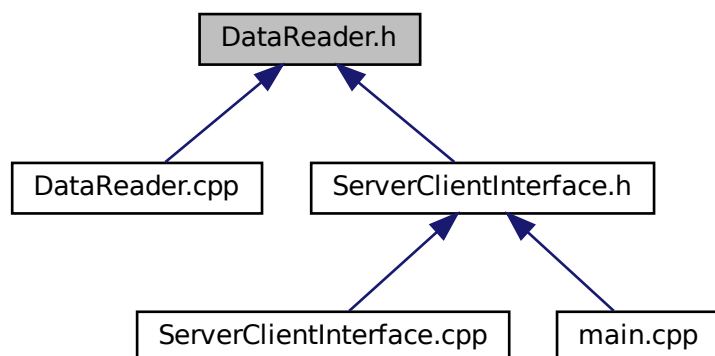
```
#include <string>
#include <vector>
#include <map>
#include <fstream>
#include <sstream>
#include <iostream>
```

```
#include "Errors.h"
```

Граф включаемых заголовочных файлов для `DataReader.h`:



Граф файлов, в которые включается этот файл:



## Классы

- class `DataReader`

Класс для получения БД клиентов

### 4.2.1 Подробное описание

Заголовочный файл для модуля `DataReader`.

Автор

Дьякова С.М.

Версия

1.0

Дата

12.12.2024

Авторство

ИБСТ ПГУ

## 4.3 DataReader.h

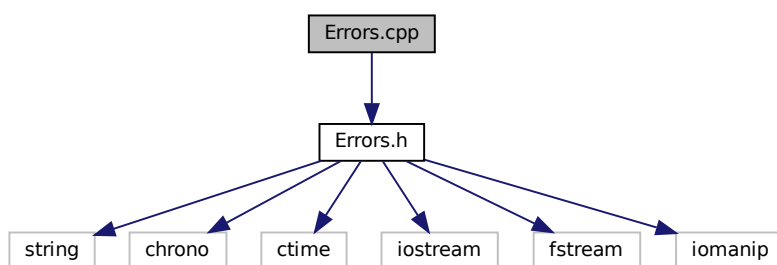
[См. документацию.](#)

```
1
9 #include <string>
10 #include <vector>
11 #include <map>
12 #include <fstream>
13 #include <sstream>
14 #include <iostream>
15
16 #include "Errors.h"
17
18 using namespace std;
19
25 class DataReader{
26     public:
27
28         string get_FileReader();
29         void set_FileReader(string file);
30
31         pair<vector<string>, vector<string>> getClient();
32     private:
33         string FileReader;
34         Errors Err;
35 };
```

## 4.4 Файл Errors.cpp

```
#include "Errors.h"
```

Граф включаемых заголовочных файлов для Errors.cpp:

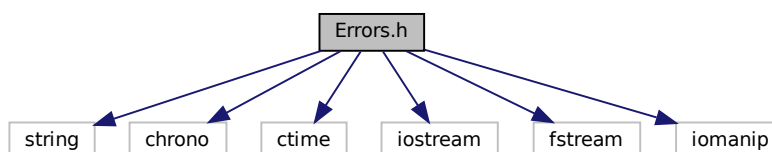


## 4.5 Файл Errors.h

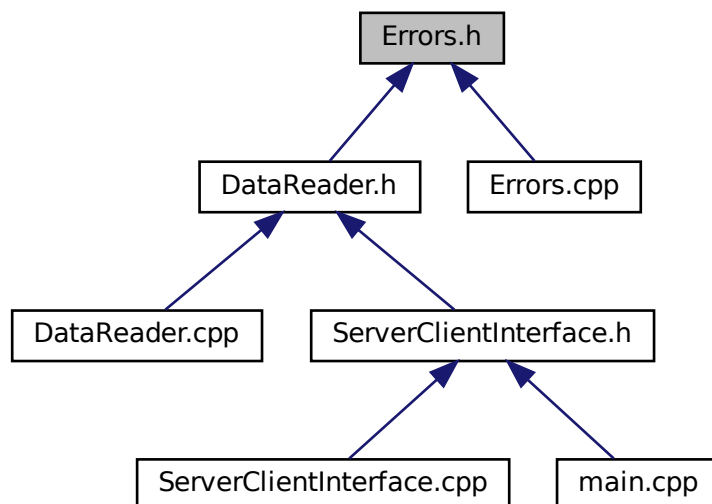
Заголовочный файл для модуля [Errors](#).

```
#include <string>
#include <chrono>
#include <ctime>
#include <iostream>
#include <fstream>
#include <iomanip>
```

Граф включаемых заголовочных файлов для Errors.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [Errors](#)

Класс обработки ошибок

### 4.5.1 Подробное описание

Заголовочный файл для модуля [Errors](#).

Автор

Дьякова С.М.

Версия

1.0

Дата

12.12.2024

Авторство

ИБСТ ПГУ

## 4.6 Errors.h

[См. документацию.](#)

```

1
9 #include <string>
10 #include <chrono>
11 #include <ctime>
12 #include <iostream>
13 #include <fstream>
14 #include <iomanip>
15
16 using namespace std;
17
22 class Errors{
23     public:
24         string get_File_Log();
25         void set_File_Log(string file);
26
27         void error_recording(string flag, string info);
28
29     private:
30         string File_Log;
31 };

```

## 4.7 Файл main.cpp

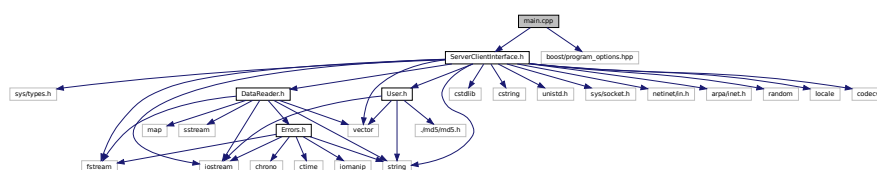
Функция для предоставления пользователю справки по использованию программы

```

#include "ServerClientInterface.h"
#include <boost/program_options.hpp>

```

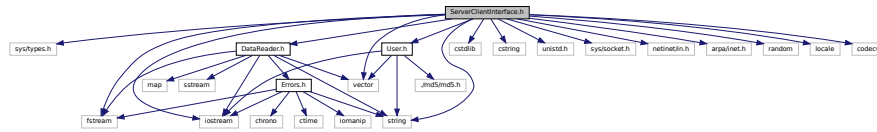
Граф включаемых заголовочных файлов для main.cpp:



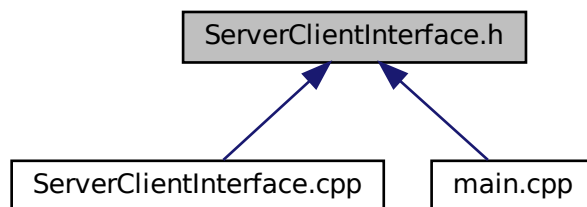


```
#include <codecvt>
#include "DataReader.h"
#include "User.h"
```

Граф включаемых заголовочных файлов для ServerClientInterface.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [ServerClientInterface](#)  
Класс для взаимодействия сервера с клиентами

### 4.9.1 Подробное описание

Заголовочный файл для модуля [ServerClientInterface](#).

Автор

Дьякова С.М.

Версия

1.0

Дата

12.12.2024

Авторство

ИБСТ ПГУ

## 4.10 ServerClientInterface.h

См. документацию.

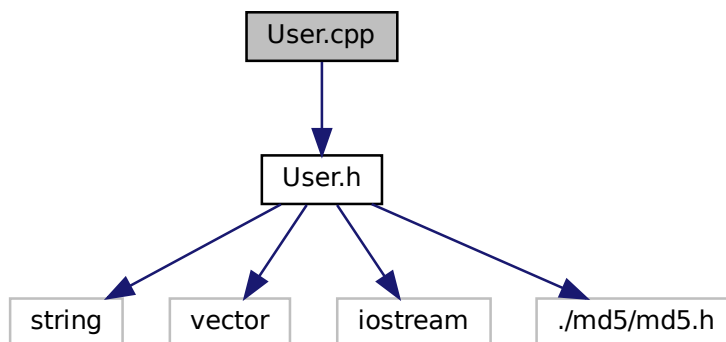
```
1
9 #pragma once
10 #include <sys/types.h>
11 #include <iostream>
12 #include <cstdlib>
13 #include <cstring>
14 #include <unistd.h>
15 #include <sys/socket.h>
16 #include <netinet/in.h>
17 #include <arpa/inet.h>
18 #include <fstream>
19 #include <vector>
20 #include <random>
21
22 #include <string>
23 #include <locale>
24 #include <codecvt>
25
26 #include "DataReader.h"
27 #include "User.h"
28
29 using namespace std;
30
31 class ServerClientInterface{
32 public:
33     int interaction(string database, string logFile);
34     uint64_t sum_vec();
35
36     string get_address();
37     void set_address(string address1);
38
39     int get_port();
40     void set_port(int port1);
41
42     vector<uint64_t> get_vec();
43     void set_vec(vector<uint64_t> v);
44
45     string get_salt();
46     void set_salt();
47
48     pair<vector<string>, vector<string>> get_DB_clients();
49     void set_DB_clients(vector<string> login, vector<string> password);
50
51 private:
52     string address;
53     int port;
54     vector<uint64_t> vec;
55     string salt;
56     pair<vector<string>, vector<string>>DB_clients;
57     Errors Err;
58 };
59 }
```



## 4.11 Файл User.cpp

```
#include "User.h"
```

Граф включаемых заголовочных файлов для User.cpp:

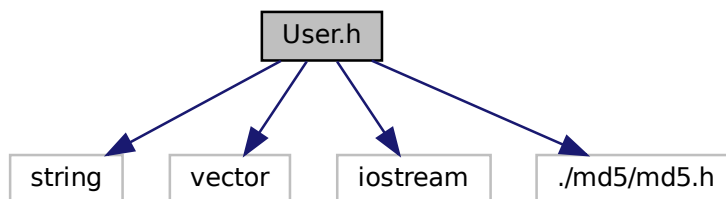


## 4.12 Файл User.h

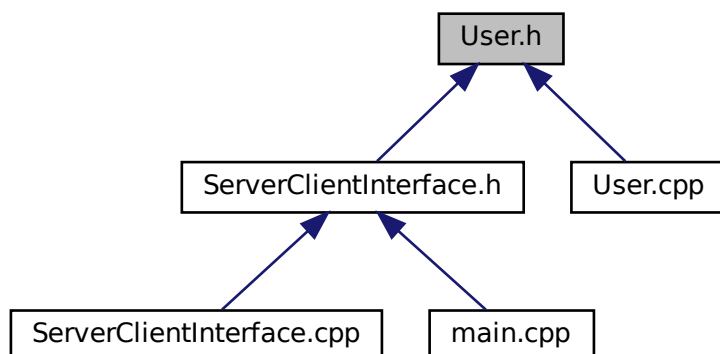
Заголовочный файл для модуля [User](#).

```
#include <string>
#include <vector>
#include <iostream>
#include "./md5/md5.h"
```

Граф включаемых заголовочных файлов для User.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [User](#)

Класс, представляющий информацию о подключенном пользователе

### 4.12.1 Подробное описание

Заголовочный файл для модуля [User](#).

Автор

Дьякова С.М.

Версия

1.0

Дата

09.12.2024

Авторство

ИБСТ ПГУ

## 4.13 User.h

[См. документацию.](#)

```
1
9 #include <string>
10 #include <vector>
11 #include <iostream>
12
13 #include "../md5/md5.h"
14
15 using namespace std;
16
22 class User{
23     private:
24         string ID;
25         string hash;
26     public:
27         bool CheckLogin(vector<string> Db_ID);
28         bool CheckPassword(vector<string> Db_password, vector<string> Db_ID, string SALT, string SendHash);
29
30         string get_ID();
31         void set_ID(string ID1);
32
33         string get_hash();
34         void set_hash(string hash1);
35
36 };
```



# Предметный указатель

CheckPassword  
    User, [11](#)

DataReader, [5](#)  
    getClient, [5](#)  
DataReader.cpp, [13](#)  
DataReader.h, [13](#)  
DataReader\_fix, [6](#)

error\_recording  
    Errors, [7](#)  
Errors, [6](#)  
    error\_recording, [7](#)  
Errors.cpp, [15](#)  
Errors.h, [16](#)  
Errors\_fix, [8](#)

getClient  
    DataReader, [5](#)

main.cpp, [17](#)

Result, [8](#)

Server\_fix, [9](#)  
ServerClientInterface, [9](#)  
    sum\_vec, [10](#)  
ServerClientInterface.cpp, [18](#)  
ServerClientInterface.h, [18](#)  
sum\_vec  
    ServerClientInterface, [10](#)

User, [10](#)  
    CheckPassword, [11](#)  
    CheckLogin, [11](#)  
User.cpp, [21](#)  
User.h, [21](#)  
User\_fix, [12](#)

CheckLogin  
    User, [11](#)