**Contents**

[1 Introduction 2](#_Toc370402491)

[1.1 Client Identification 2](#_Toc370402492)

[1.2 Current System 2](#_Toc370402493)

[1.3 Problems 2](#_Toc370402494)

[1.4 Section Appendix 2](#_Toc370402495)

[2 Investigation 3](#_Toc370402496)

[2.1 The current system 3](#_Toc370402497)

[2.1.1 Data sources and destinations 3](#_Toc370402498)

[2.1.2 Algorithms 3](#_Toc370402499)

[2.1.3 Data flow diagram 5](#_Toc370402500)

[2.1.4 Input Forms, Output Forms, Report Formats 5](#_Toc370402501)

[2.2 The Proposed System 5](#_Toc370402502)

[2.2.1 Data sources and destinations 5](#_Toc370402503)

[2.2.2 Data flow diagram 6](#_Toc370402504)

[2.2.3 Data dictionary 6](#_Toc370402505)

[2.2.4 Volumetric 7](#_Toc370402506)

[3 Objectives 7](#_Toc370402507)

[3.1 General Objectives 7](#_Toc370402508)

[3.2 Specific Objectives 7](#_Toc370402509)

[3.3 Core Objectives 7](#_Toc370402510)

[3.4 Other Objectives 7](#_Toc370402511)

[4 E-R Diagrams and Descriptions 7](#_Toc370402512)

[4.1 E-R Diagram 8](#_Toc370402513)

[4.2 Entity Descriptions 8](#_Toc370402514)

[5 Object Analysis 8](#_Toc370402515)

[5.1 Object Listing 8](#_Toc370402516)

[5.2 Relationship diagrams 9](#_Toc370402517)

[5.3 Class definitions 9](#_Toc370402518)

[5.3.1 Client/AuthDevice 9](#_Toc370402519)

[5.3.2 Server 10](#_Toc370402520)

[5.3.3 Records 10](#_Toc370402521)

[6 Other Abstractions 10](#_Toc370402522)

[6.1 Graphs 10](#_Toc370402523)

[7 Constraints 11](#_Toc370402524)

[7.1 Hardware 11](#_Toc370402525)

[7.2 Software 11](#_Toc370402526)

[7.3 Time 11](#_Toc370402527)

[7.4 User Knowledge 11](#_Toc370402528)

[7.5 Access restrictions 11](#_Toc370402529)

[8 Limitations 12](#_Toc370402530)

[8.1 Areas which will not be included in computerisation 12](#_Toc370402531)

[8.2 Areas considered for future computerisation 12](#_Toc370402532)

[9 Solutions 12](#_Toc370402533)

[9.1 Alternative solutions 12](#_Toc370402534)

[9.1.1 Face Recognition 12](#_Toc370402535)

[9.1.2 Other biometrics 12](#_Toc370402536)

[9.1.3 NFC 12](#_Toc370402537)

[9.2 Justification of chosen solution 12](#_Toc370402538)

Analysis

# Introduction

## Client Identification

The client is Paul Milne and this system will be for his household, the interest in developing this better system is purely for ease of use.

## Current System

The alarm system is for securing the house, there are alarms situated around the house on windows and doors which are wirelessly connected to the local network. The purpose for the alarm system is simply security.

Currently, to disable or enable the alarm a user must type in the set PIN into a number pad which is located near the stairs in the house. This is then sent to the server which controls all of the alarms in the house, the packet of data contains information which authenticates the device and the PIN which has been sent. If the PIN is correct, data is sent back to the number to give feedback stating the alarms have been now enabled or disabled. Otherwise it will reply with access denied and allow three more attempts before locking itself until a master password is inputted.

## Problems

The main problem with the current system is that it is just too much hassle to keep track of a changing PIN and even to just enter it in twice a day. Replacing the current system with something that takes just a touch or even a look would save a lot of hassle with remembering PINs and setting them. This also leads onto the security itself of the PIN, it can easily be found on a piece of paper mistakenly thrown away which could give anyone access to the house.

## Section Appendix

**Interview transcript:**

[BEGIN INTERVIEW]

[INTERVIEWER: HARRY MILNE]

[INTERVIEWEE: PAUL MILNE]

[HARRY: "What is your current system?"]

[PAUL: "I currently have an alarm system in my house which is turned on and off via a number pad"]

[HARRY: "So, how does the current alarm system work?"]

[PAUL: "The number pad is connected via Ethernet to the server in my study, this server then controls the alarms around the house"]

[HARRY: "What are the specifications of the server?"

[PAUL "i7 4th-Generation processor, 16GB of RAM and a 4 TB RAID 0 HDD setup running on Debian"]

[HARRY: "OK, so what is the problem or problems with the current system?"]

[PAUL: "I think that there should be a better authentication method for turning the alarms on and off"]

[HARRY: "Does the current system log access?"]

[PAUL: "No, but I think that would be useful information to have"]

[HARRY: "Does the current system have any output of data?"]

[PAUL: "No, the system uses a PIN which we change every 6 months"]

[HARRY: "How many users would be needing access?"]

[PAUL: "2"]

[HARRY: "Do you have a budget?"]

[PAUL: "Yes, I'd only like to spend around 50 GBP"]

[HARRY: "Do you have any pre-existing ideas about the solution?"]

[PAUL: "Some kind of biometric solution would be viable"]

[HARRY: "Thank you for your time, I will be in contact soon"]

[PAUL: "Thanks"]

[END INTERVIEW]

# Investigation

## The current system

### Data sources and destinations

|  |  |  |  |
| --- | --- | --- | --- |
| **Source** | **Data** | **Example Data** | **Destination** |
| Client | Combination of Integers (length varying from 4 to 8 digits) | 1234 | Numpad |
| Numpad | Packet containing combination and authentication data. | ($auth\_info, “1234”) | Server |
| Server | Packet containing success status and authentication data. | ($auth\_info, True) | Numpad |
| Server | Packet containing authentication data and state that the device should be in. | ($auth\_info, False) | Alarm |

### Algorithms

#### Number pad Process

*WHILE True*

*key\_entered ← NoneType*

*pin ← “”*

*WHILE key\_entered ≠ enter*

*key\_entered ← USERINPUT*

*pin ← pin + key\_entered*

*ENDWHILE*

*IF LEN(pin) ≠ 4 THEN*

*OUTPUT “ERR: LENGTH”*

*CONTINUE*

*server.send\_packet(pin) #packet sent to server*

*response ← server.recv() #packet response from server (bool)*

*IF response THEN*

*OUTPUT “GRANTED”*

*ELSE*

*OUTPUT “DENIED”*

*ENDIF*

*ENDWHILE*

#### Server Process

*WHILE True*

*network.listen() #listen to packets being sent*

*data ← network.recv() #retrieve data from received packet*

*IF data matches valid packet THEN*

*IF data passes authentication THEN*

*IF user in data matches database THEN*

*network.send(TRUE) #send positive response back to device*

*ELSE*

*network.send(FALSE) #user did not match database*

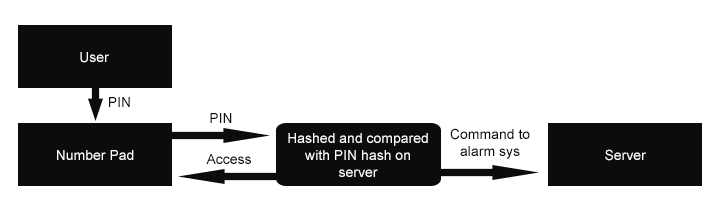
*ENDIF*

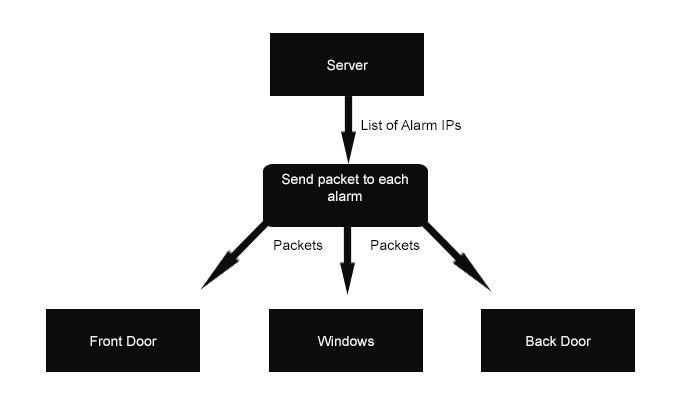
*ENDIF*

*ENDIF*

*ENDWHILE*

### Data flow diagram





### Input Forms, Output Forms, Report Formats

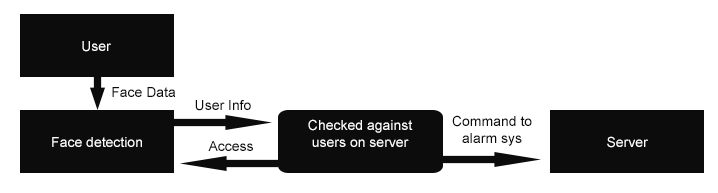
The only input to the current system is via the number pad to set and enter the PIN each time.

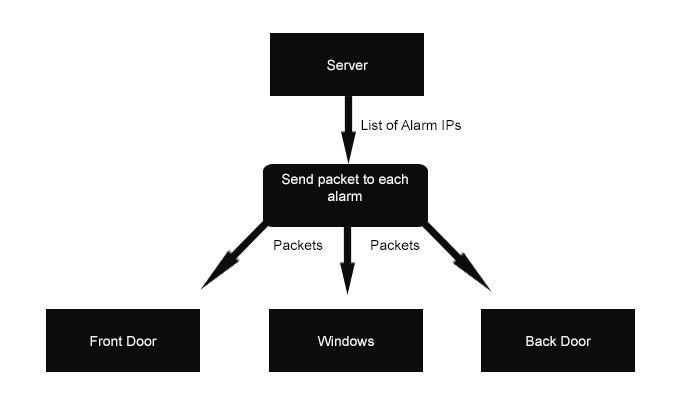
## The Proposed System

### Data sources and destinations

|  |  |  |  |
| --- | --- | --- | --- |
| **Source** | **Data** | **Example Data** | **Destination** |
| Client | Face recognition data | This will be binary data. | Auth Device |
| Auth Device | Packet containing who the user is and the authentication data. | ($auth\_info, “Paul”) | Server |
| Server | Packet containing success status and authentication data. | ($auth\_info, True) | Auth Device |
| Server | Packet containing authentication data and state that the device should be in. | ($auth\_info, False) | Alarm |

### Data flow diagram





### Data dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Length** | **Validation** | **Example Data** | **Comment** |
| UserID | Integer | 12 bit | N/A | 4095 | Unique to each user |
| UserName | String | 20 chars | Alphanumeric and <= 20 characters | “Paul” | Stored as tuple, first name index 0. |
| UserFaceData | Binary | N/A | N/A | N/A | Face recognition data. |
| EntryDate | String | 10 chars | N/A | “01/01/2013” | Generated when entry is validated. |
| Access\_Granted | Boolean | 1 bit | N/A | 1 | True if user was granted access. |
| Auth\_List | List | Variable | Valid IPs | [“192.168.0.10”,  “192.168.0.11”) | A list of saved authenticated devices |
| Auth\_Data | Binary | 1024-bit | N/A | (RSA Keys) | Used to check device’s identity |

### Volumetric

The size of the RAID that the database will be stored on is 4TB, this is populated with some other data but mostly is free space, so therefore the amount of logs or users will never feasibly reach over the size of the server’s HDD because this is a home implementation of the system so there are minimal users.

# Objectives

## General Objectives

* Quick and easy to use method of user verification
* Easy additional user setup
* Communication with the server over the network
* Store users and access logs in a SQL database

## Specific Objectives

**User Verification**

* Ability to recognize faces
* Ability to tell different faces from others
* Must be able to match a name to that face

**User Setup**

* UI to edit each user or add users
* Power user tools

**Network Communication**

* Access to local area network
* Receive commands and send commands
* Secure

**SQL Database**

* Table containing each user
* Access logs with a relationship to a user
* Secure interface with communicating to the database

## Core Objectives

* User verification via face recognition
* Interface with the server
* Server to turn the alarms on and off

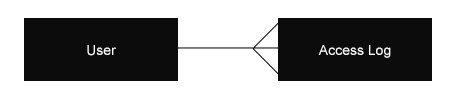
## Other Objectives

* Socket encryption
* Control over what time slots people are allowed to turn alarms off
* Ability to edit users from different computers

# E-R Diagrams and Descriptions

## E-R Diagram

**Database Relationships**



**Hardware Relationship**

****

## Entity Descriptions

**User**(user\_no, user\_name, face\_data)

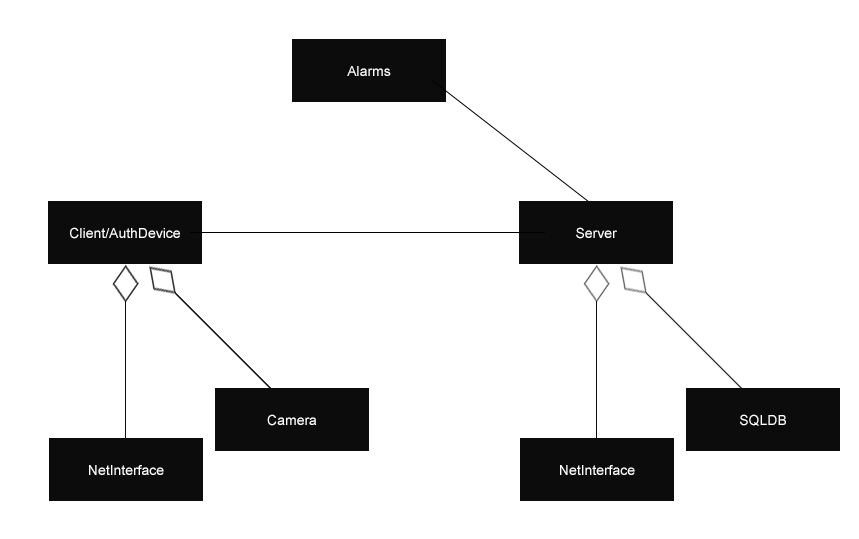
**Access**(access\_no, date, *user\_no*)

# Object Analysis

## Object Listing

* AuthDevice
  + Camera
  + NetworkInterface
* Server
  + SQLDatabase
  + NetworkInterface
* Records
  + UserRecord
  + AccessRecord

## Relationship diagrams



## Class definitions

### Client/AuthDevice

#### KEY

|  |
| --- |
| NAME |
| ATTRIBUTES |
| METHODS |

#### Since this will be coded in Python methods aren’t needed to retrieve attributes.

|  |
| --- |
| **Camera** |
| active |
| resolution |
| face\_detected |
| change\_res |
| scan |
| log\_user |

|  |
| --- |
| **NetworkInterface** |
| local\_address |
| master\_address |
| master\_port |
| update |
| auth |
| send\_record |
| recv\_record |

### Server

|  |
| --- |
| **SQLDatabase** |
| size |
| number\_of\_users |
| number\_of\_records |
| search |
| add\_record |
| rm\_record |
| query |

|  |
| --- |
| **NetworkInterface** |
| local\_address |
| local\_port |
| authed\_devices |
| auth |
| listen |
| reply |
| process |

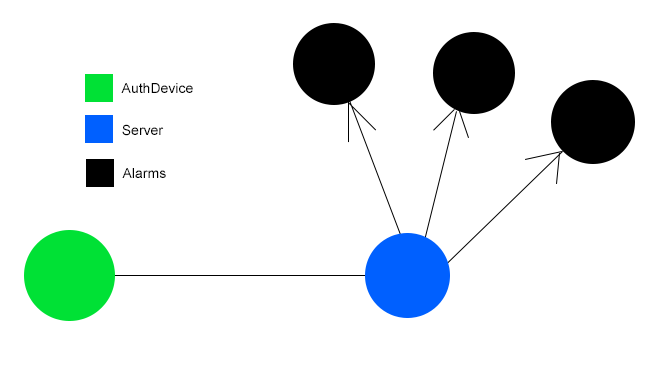
### Records

|  |
| --- |
| **UserRecord** |
| user\_no |
| user\_name |
| user\_facedata |

|  |
| --- |
| **AccessRecord** |
| date |
| user |
| user\_no |

# Other Abstractions

## Graphs



# Constraints

## Hardware

The budget allows the purchase of a Raspberry Pi and camera; this allows the project to be very flexible as there aren’t many coding constraints. The server hardware will have no issue dealing with the amount of users (2) in a household, and logging the access of them.

This helps with development as it means I don’t have restrictions in how I code or what libraries I choose to use meaning a lot more rapid development.

## Software

There are no constraints on what software to use other than the server’s operating system being a Linux distribution, which doesn’t really impact me since I won’t have to install anything on the server apart from Python 3 and a SQL database, and they both work with all Linux distributions.

This helps with development as it means I don’t have restrictions in how I code or what libraries I choose to use meaning a lot more rapid development.

## Time

The deadline for the project is set by the college which is Monday the 17th of February, 2014. The user has expressed no need for the system to be completed before this time.

## User Knowledge

The IT knowledge of this user is very high, so this means I will be able to explain exactly what I am implementing to him which just speeds up the process. This means I can implement the UI with efficiency of use in mind and because of his high IT proficiency he will be able to use it.

## Access restrictions

Admins will be able to access the data log and user management on the server; users will only be interacting with the mounted authentication device. This means I won’t have to implement as much protection from accidental data deletion from users who don’t know what they are doing; since they won’t have access to it.

# Limitations

## Areas which will not be included in computerisation

* Other biometrics

Other biometrics such as finger print or retina recognition would improve the accuracy and security of this system but the hardware needed to implement this exceeds the client’s budget.

## Areas considered for future computerisation

* Web interface/apps

Forms of modifying the system via either the Internet or other local devices while not being sat in front of the server or authentication device would be a good addition to the system; this allows for more user flexibility.

# Solutions

## Alternative solutions

### Face Recognition

Face recognition is a method of verification which allows no physical interaction from the user. This method of authentication has also been used by other companies such as Google’s Android operating system, in which you glance or smile at your phone and it allows you access. This means there will already be plethora of documentation and information that can be found; this could help my development because it means a lot of testing has been done for me.

### Other biometrics

There are a few options with biometrics, including fingerprint, iris scanning, voice recognition; most of these methods are very much viable for user authentication but at an expense; they cost more. Most of these could be implemented into the future of the project because they add layers of security.

### NFC

The amount of devices capable of NFC (near field communication) has recently grown substantially, so this makes NFC a possible solution. However, this still isn’t the majority of devices so a lot of users will not be able to use their personal devices for this method. Yet, if this system was being designed for a big company, they might distribute a company phone which could have this as one of its features, but the industry seems to be going in the opposite direction; the push is towards the use of personal devices for work.

## Justification of chosen solution

I decided to choose face recognition as the method of authentication because of the low budget and low costs of obtaining the equipment needed to use face recognition. All you require is a camera and the software will do the rest of the work. This method also detaches the user needing any kind of tech knowledge, all they need to do is look at the device and as soon as they are recognized they have done what they need to do; nothing complex to teach.