



<UNDECIDABLES>

COSBAS User Manual

Git: <https://github.com/undecidables/Documentation>

GitHub Organisation: <https://github.com/undecidables>

The Team:

Elzahn Botha *13033922*
Jason Richard Evans *13032608*
Renette Ros *13007557*
Szymon Ziolkowski *12007367*
Tienie Pritchard *12056741*
Vivian Venter *13238435*

August 2015

Contents

1	Introduction	2
2	System Overview	2
2.1	Overview Description:	2
2.2	Pinpoint Descriptions:	2
2.2.1	COSBAS-Client	2
2.2.2	Web-Client	2
2.2.3	Bookings and Appointments	2
2.2.4	Temporary Access	2
3	System Configuration	3
3.1	Graphical System Configuration Diagram	3
3.2	Description of the Equipment used as Illustrated on the Diagram	3
3.3	System Configuration Explained	4
3.4	Communication and Networking	4
3.5	New Biometric Types	4
3.5.1	Server Side	4
3.5.2	Client Side	4
4	Installation of the COSBAS System	5
4.1	Obtaining the Software	5
4.2	Installing the Server Component	5
4.2.1	Configuration	5
4.2.2	Building	6
4.3	Installing the Client Component	6
4.3.1	Configuration	6
4.3.2	Building	6
5	Getting Started	6
5.1	Getting Access to the System	6
5.2	Register on the System	6
5.3	Change of Login Details	7
5.4	General Walkthrough of the System	7
5.4.1	COSBAS-Client	7
5.4.2	Web-Client	7
5.5	Exit the System	8
6	Using the System	8
6.1	Using the Appointment System	8
7	Troubleshooting	10

1 Introduction

The COSBAS system is a highly secure and modern access control system that uses Biometric data to authorize the individuals request to enter the department and offices. This document will specify how to place the COSBAS system in a working state. The document instructs on the necessary steps to install the hardware needed by our system as well as installing the COSBAS system software on the relevant computers.

2 System Overview

2.1 Overview Description:

The COSBAS (Computer Science Biometric Access System) is a secure system that uses Biometric inputs (such as facial recognition and fingerprint scanning) to unlock and gain access to the department and offices.

2.2 Pinpoint Descriptions:

2.2.1 COSBAS-Client

The COSBAS client is the hardware aspect of the COSBAS system. It will allow users to capture biometric data to be used as authentication for access to the department. The images are taken (be it facial, fingerprint etc.) and then sent off to the server for authentication while waiting for a response from the server to permit the access or not. Initially the COSBAS system will only have the functionality of Facial Recognition and Fingerprint Identification. With facial recognition, a set of images are take but only the images with the most centered face is sent to the server for authentication purposes. If no face is detected, then the user would have to retry the process. Fingerprint images are close proximity images and hence will always be accurate enough to send it directly to the server for authentication.

2.2.2 Web-Client

The interface for the COSBAS system will be a responsive webpage the user (authorized and temporary visitors) may use to request permission for access to the department. They can also book appointments with members of faculty on the system.

2.2.3 Bookings and Appointments

An unauthorized user can book an appointment with an employee enrolled in the system by making use of a Calendar integration feature on the web based interface. Authorized users can then either accept or decline the booking for an appointment of which the person whom made the booking is notified of the status of their booking via an email.

2.2.4 Temporary Access

Once a booking has been accepted by the authorized user of COSBAS, the relevant users will be notified via email containing a link that will expose their temporary access code generated by the COSBAS system. There will also be another link in the email the user may click on

to cancel the appointment with the associated COSBAS user. In such a case, the temporary access code will be revoked.

3 System Configuration

3.1 Graphical System Configuration Diagram



3.2 Description of the Equipment used as Illustrated on the Diagram

- **Client - Raspberry PI:** Is a very small computer with a very low power consumption. The PI can handle quite a few input/output devices via the USB/HDMI/LAN/GPIO ports.
- **Camera:** A device which will capture an image of the user that will want to authenticate via biometrics.
- **Fingerprint scanner:** A device that will capture the fingerprint of the user which will be used for authentication.
- **Keypad:** Will be used by users that will gain access to the building via the keycode.
- **USB Hub:** This device will allow the system to connect more than one device via USB to the client as well as give power to the client.
- **Pressure mat:** This device will allow the system to pick up that the user is ready to be authenticated (usually facial recognition) to gain access to the building.

- **Electromagnet door lock:** Will keep the door locked until the client has successfully authenticated the user.
- **Server:** Will be used for all the heavy computations such as facial/finger print recognition, etc.

3.3 System Configuration Explained

The system is made up of pluggable authentication devices such as a keypad, camera and fingerprint scanner, as well a client (Raspberry Pi) and a server. The entire authentication process is started as soon as the user steps onto the pressure mat sending the authentication data to the Raspberry Pi for processing before it is sent to the server for authentication. Once authentication is complete on the server, a reply will be sent to the client and the client will act accordingly.

3.4 Communication and Networking

The pressure mat, USB Hub and electromagnet door lock all get their power from the main outlet while the keypad, camera and fingerprint scanner will be getting their power via the USB connection. The Raspberry Pi also connects to the USB Hub both for power and data transfer between the authentication devices and the client. The client communicates with the server via a LAN cable and the pressure mat and electromagnet door lock connects to the client GPIO pins via I/O cables.

Once the user steps onto the pressure mat a signal is sent to the client which gets its data from the authentication devices and processes that data before sending it off to the server if it was valid data. The server authenticates the person and sends the data back to the client that will then, if the authentication was successful, open the door for the user.

3.5 New Biometric Types

3.5.1 Server Side

Adding a new biometric validator to the system requires source code modification.

1. **Write the validator**

Each validator should extend the abstract `AccessValidator` class in the `cosbas.biometric.validators` package.

2. **Define it as a bean**

Declare the class a bean by adding the Spring `@Component` annotation to it.

3. **Register it on the system**

To register the biometric type on the system add it to the `cosbas.biometric.biometricTypes` enum with its validator class. Eg. `CODE (CodeValidator.class)`. The type should be uppercase.

3.5.2 Client Side

To be documented.

4 Installation of the COSBAS System

The COSBAS System consists of two main components:

- The access and appointment server
- The access client

Due to the nature of the system the average user who just wants to use the appointment or access system that is already in place can ignore this section.

The appointment system can be accessed through a normal web browser and gaining access after you have been registered on the system should be as simple as standing in front of the door and entering an access code or activating the biometric devices.

4.1 Obtaining the Software

The system's Java source code as well as all related documentation can be found on the Undecidables GitHub organisation at <https://github.com/undecidables>

The important repositories in this organisation:

- **Documentation:** The documentation repository is home to the project's Wiki and also contains the Functional Requirements, Architectural Requirements and User Manual.
- **COSBAS-Server:** This repository contains the server component of the project. It consists of Java sourcecode, Thymeleaf view templates, a Gradle build file and a few configuration files.
- **COSBAS-Client:** This program is the client application to request access through the biometric system. It is also written in Java and uses the Gradle build system.

The following two repositories are less important to the end user, but might give some insight into the beginning of the system's development:

- **Research:** This repository was created as a central location for the reasearch conducted at the beginning of the project, especially reasearch about hardware, technologies and frameworks.
- **Tenders:** This repository contains the tender documents the team created for the original COS301 project proposals. It is not important to a user of the COSBAS system.

4.2 Installing the Server Component

4.2.1 Configuration

Common system properties such as the server port, the LDAP address and the mongoDB address can be set in the application.properties file located in 'src/main/resources'. The so-called 'secret' file needed to use the Google Calendar API should also be placed in this location.

4.2.2 Building

We use the Gradle build system to manage dependencies:

- On a system that has Gradle 2.3 installed simply run 'gradle build' to create an executable jar and use 'gradle run' to execute it.
- On a system that does not have at least Gradle 2.3 installed, the gradle wrapper (that is on the repository) can be used. Simply use 'gradlew' instead of 'gradle' when you are in the project's root directory. (On a Linux system maybe './gradlew'). This wrapper will then download the correct version of Gradle and use it to build the project.
- gradle and gradlew might require super user or administrator privileges.

4.3 Installing the Client Component

The client component should ideally be deployed on a small computer like a Raspberry Pi located at a door or some other access channel. The hardware configuration for the client was discussed in section 3 in this document.

4.3.1 Configuration

Similarly to the structure used in the server component common application properties (eg. where the door it is located at and the server's address) can be set in the config.properties file located in src/main/resources.

4.3.2 Building

The client also uses the Gradle build system so it can be built and executed in the same way as the server (see section 4.2.2).

Due to the minimal resources of the Raspberry Pi's and to save some time it is recommended that the client is built and set up on one Pi and then cloned to the others. Small configuration changes can be made to each Pi at a later stage.

5 Getting Started

5.1 Getting Access to the System

To gain access to the COSBAS System through the web client you need the following:

- **Username** - This username is the same as the username needed to login to the CS Website. Usually it is the employee number of the staff member.
- **Password** - This is the password that you use to login to the CS Website.

5.2 Register on the System

If you need to register on the system you need to go to the department where they will add you to their LDAP servers such that you can login to the system.

Note: Only staff members or frequent recognised members will be able to get access to the system by means of the CS username and password. If you are not such a member then you can view the web client as an guest and still make appointments as you wish.

5.3 Change of Login Details

The COSBAS System will not be able to change your username or password. To change your CS login details, which is your COSBAS login details, you need to go to the department since the COSBAS System authenticates the user through the LDAP server of the University of Pretoria.

5.4 General Walkthrough of the System

5.4.1 COSBAS-Client

As mentioned in section 2.2.1, the client consists of the hardware, which is the Camera, Fingerprint Scanner, Raspberry Pi, Keypad and Pressure Mat. The client will do the biometric detection on the Raspberry Pi and will send the necessary data to the server for the biometric authentication.

Walkthrough per Biometric/Authentication Method,

- **Facial Recognition** - Stand on the pressure pad to initialize the camera to take a photo. After an photo has been taken facial detection will occur in the Raspberry Pi to detect if there is in fact a face in the image. When a face has been detected by the client, the image is send to the server for authentication where the user can gain access to the department if successful authentication has been the case.
- **Fingerprint Scanning** - Place a finger on the fingerprint scanner. Either one of the following fingers may be used,
 - Left Thumb
 - Left Index Finger
 - Right Thumb
 - Right Index FingerThe fingerprint scan will be authenticated against current stored copies of the user's fingerprints. The user will gain access if the authentication has been successful.
- **Authentication Key** - Enter the authentication key on the keypad.
 - For **registered users** such as the staff members this will be the dedicated authentication key you will be provided with once registration for the COSBAS System has been done.
 - For **temporary/guest users** this will be the key that was provided to you via email after the appointment has been approved by the particular staff member.

5.4.2 Web-Client

- **Registered User**
 - Login with your COSBAS Login Details (see section 5.1).

- If you have a gmail account and would like to link your Google Calendar to the appointment system, grant permission to Google to get access to your Calendar (You will be redirected to the grant access page of Google).
- Go to the appointment page to approve or decline appointments.
- Go to the booking page to make a booking by using the online form.

- **Guest User**

- No login will be needed.
- Go to the view calendar page to view the calendar of a specific staff member.
- Go to the booking page to make a booking by using the online form.

- **Admin User**

- Login with your COSBAS Login Details (see section 5.1)
 - Go to the appointment page to approve or decline appointments.
 - Go to the booking page to make a booking by using the online form.
 - Go to the registration page to upload the biometric data of a specific staff member, that is to register the user on the COSBAS System.
- Note:** When registering a user that user will need to provide his/her login details (CS Login Details) before registration can be done.

5.5 Exit the System

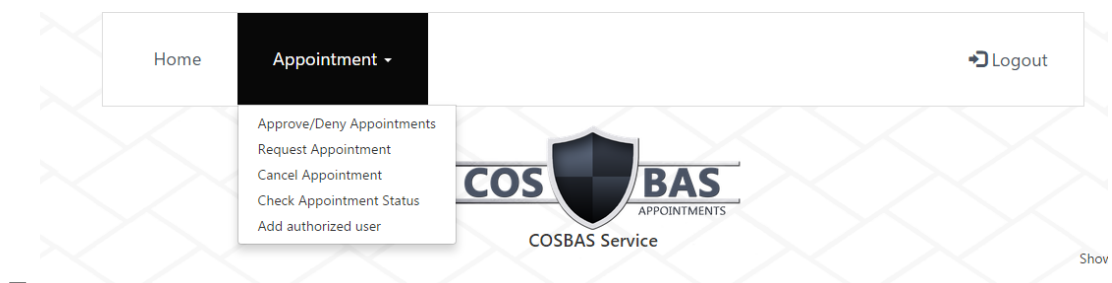
To exit the system depends on the type of user you are,

- if you are a **registered/admin user** you can simply click on the logout button to exit the system.
- if you are a **guest user** you can exit the system by simply closing the browser.

6 Using the System

6.1 Using the Appointment System

- Navigation



- The navigation is simple to use. It consists of 5 pages which a user can navigate to and use. These pages are:
 - * Approve/Deny Appointments
 - This page can be used to view pending appointments and approve or deny them.

- * Request Appointment
 - A appointment requester will use this page to request an appointment with a lecturer.
- * Cancel Appointment
 - An appointment requester or lecturer can use this page to cancel an appointment.
- * Check Appointment Status
 - A appointment requester will use this page to check if their appointment was approved by a given lecturer.

- Login


- This page is used to provide authentication to the system. The user must enter their EMPLID and password, which is authenticated with the CS department's LDAP system.


- Request Appointment

The screenshot displays the COS BAS APPOINTMENTS web interface. At the top, there is a navigation bar with a 'Home' button and an 'Appointment' dropdown menu. A 'Logout' link is located in the top right corner. The main header features the COS BAS APPOINTMENTS logo, which includes a shield icon. Below the header, a dark blue button labeled 'Request an appointment' is visible. The form itself contains several input fields: a dropdown menu for 'Appointment with:' (currently set to 'Staff member'), a text field for 'Date and Time of the meeting:' (showing '29/08/2015 11:32'), a text field for 'Number of members making the appointment:' (showing '1'), a text field for 'Appointment made by (your name/team members' names):', a text field for 'Reason for appointment:', and a text field for 'Duration of appointment:' (showing '15'). A dark blue 'Request Appointment' button is positioned at the bottom of the form.

- A student can request an appointment by selecting the intended type of member from a dropdown list.
 - They can then select the date and time of the meeting.
 - They can delegate how many people will be attending the meeting.
 - They must then give the names of all the people attending the meeting.
 - They must then specify why they want the appointment
 - They must then specify how long they want the appointment to last.
- Check Appointment

[Home](#)
[Appointment ▾](#)
[Logout](#)



 Check Appointment Status

Appointment ID:


Appointment made by (your name):


Check Appointment Status

- Users can check the status of an appointment by entering the unique appointment ID which will be e-mailed to the person who made the appointment.
- They must also provide the name of the person who made the appointment.

- Cancel Appointment

[Home](#)
[Appointment ▾](#)
[Logout](#)



 Cancel an appointment

Appointment ID:

Appointment cancelled by (your name):

Cancel Appointment

- To cancel an appointment, the user must enter the appointment ID which was e-mailed to the person who requested the appointment.
- They must also enter the name of the person who requested the appointment.

7 Troubleshooting

This will be described at a later stage.