

+IISY User Guide

About IISY

IISY is an issue reporting system that uses QR-codes as the medium to deliver issues to maintenance personnel.

The general idea for the project is to integrate three components to a working solution. Purpose for it is to create functionality for reporting an issue by scanning a QR-code with a smartphone camera. Scanning a QR-code will open a webpage with a button and when it is pressed, it will send our customer an issue report regarding the entity that the QR-code refers to. Data is stored from each action and we'll integrate our database to a maintenance dashboard. The dashboard is used by the customer to browse issued alerts. It is also possible to integrate a database directly to customer-maintained application through REST API.

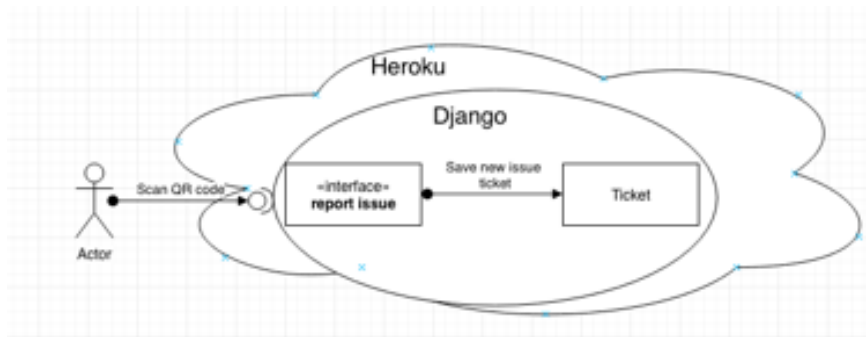
The three components for the IISY service are:

- 1) [OBJ] Smartphone native camera application is used to read a QR-code
- 2) [OBJ] Database for issues
- 3) [OBJ] Web application for viewing the issues

Features

IISY issue reporting system is built upon Django framework using Python. Django provides an easy way to integrate database and server to a working UI solution. Database is built with PostgreSQL.

Hosting is done using Heroku platform which is set up to rebuild the application whenever there is a new version in github version control.



Usage Instructions

By going to the provided URL and logging in using the provided user name and password you will be greeted with this screen

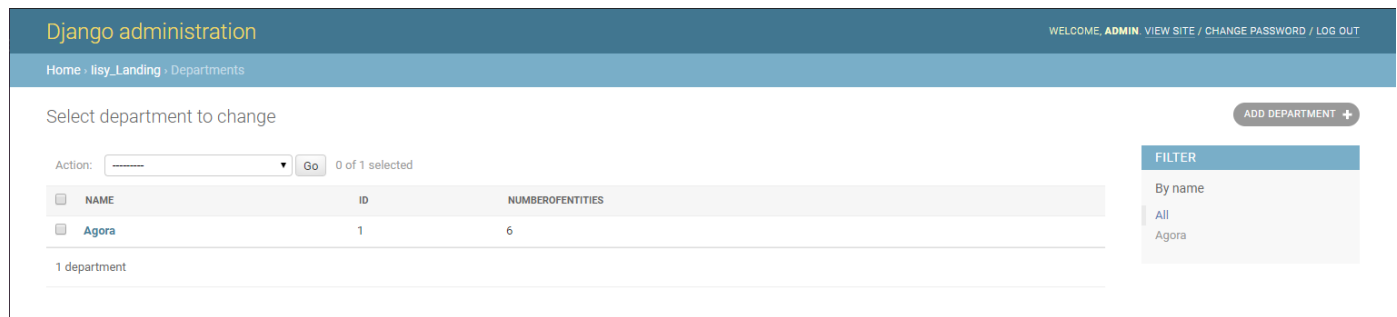
Django administration

Site administration

IISY_LANDING		
Departments	+ Add	✎ Change
Entity types	+ Add	✎ Change
Entitys	+ Add	✎ Change
Rooms	+ Add	✎ Change
Tickets	+ Add	✎ Change

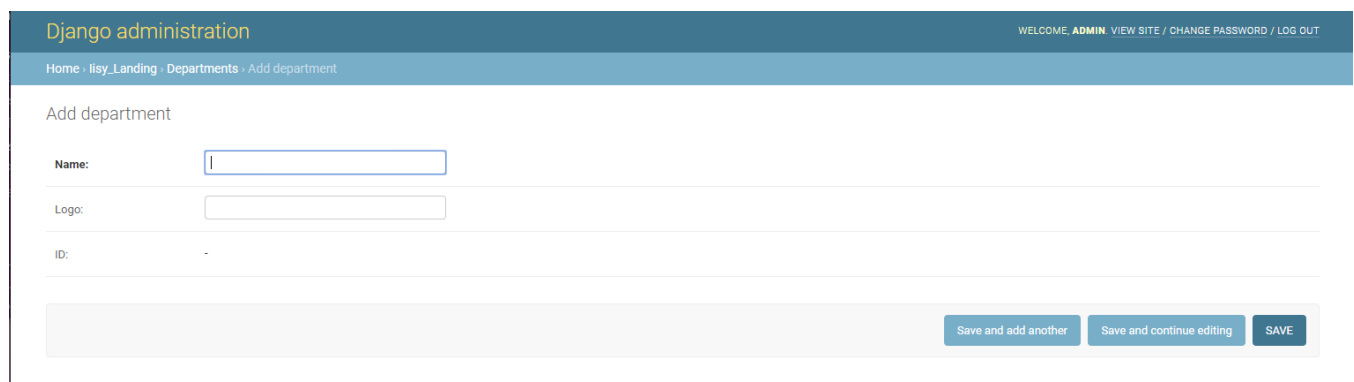
Departments

The first thing you need to do to use our system is create one [Department](#). Think of [Departments](#) as a folder containing necessary information pertaining to only that department, building, division, region, etc.



This is the [Admin panel](#) for [Departments](#). Here you can see the information on the different [Departments](#) you've created.

When pressing the [Add department button](#) you will be greeted by this screen:

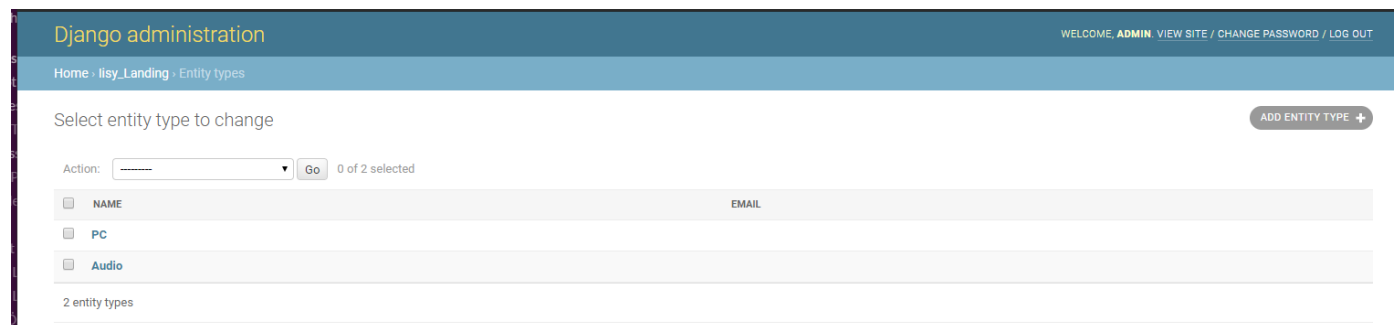


Here you enter the [Name](#) for the [Department](#) and an [URL](#) for the desired [Logo](#) you want displayed on the Issue page. After that it's created, and you can start adding [Entities](#) or [Rooms](#) to it. Only the [Name](#) is mandatory, and the [ID](#) is created automatically.

You can rename or change the logo for a desired [Department](#) by clicking on its [Name](#). You can also delete it by clicking its box and choosing [Delete selected departments](#) in the [Action list](#).

Entity type

Here is where you create your desired [Entity types](#). Think of [Entity types](#) as different presets or categories you will use when creating [Rooms](#) or [Entities](#). Entity types are not tied to a specific [Department](#).



This is the [Admin panel](#) for [Entity types](#). Here you can see the information on the different [Entity Types](#) you've created.

When pressing the [Add Entity type button](#) you will be greeted by this screen:

A screenshot of the Django administration interface showing the 'Add entity type' form. The top header bar is dark blue with 'Django administration' on the left and 'WELCOME, ADMIN VIEW SITE / CHANGE PASSWORD / LOG OUT' on the right. Below this is a light blue breadcrumb bar showing 'Home > lisy_Landing > Entity types > Add entity type'. The main content area has a heading 'Add entity type'. Below the heading are two input fields: 'Name:' and 'Email:'. At the bottom, there are three buttons: 'Save and add another', 'Save and continue editing', and 'SAVE'.

Here you enter the name for the [Entity type](#) and its [Email address](#). The [Email address](#) will be the default value used when creating [Entities](#) or [Rooms](#) if their fields are left blank. Only the [Name](#) is mandatory. The [Email](#) will be the address that receives an email when a [Ticket](#) is sent.

You can rename or change the [Email address](#) for a desired [Entity type](#) by clicking on its [Name](#) in the [Admin panel](#). You can also delete it by clicking its box and choosing [Delete selected Entity types](#) in the [Action list](#). It is possible to select multiple at once.

Entity

Here is where you create your desired [Entity](#). Think of an [Entity](#) as the device for which you want to create a [QR code](#).

Django administration

WELCOME, [ADMIN](#). [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

[Home](#) > [iisy_Landing](#) > [Entities](#)

Select entity to change

Action: Go 0 of 6 selected

<input type="checkbox"/>	ENTITYTYPE	NAME	ID	DEPARTMENT	NUMBEROFONGOINGTICKETS	LOCATION	EMAIL
<input type="checkbox"/>	PC	Windows	6	Agora	0	1	
<input type="checkbox"/>	PC	Windows	5	Agora	0	1	
<input type="checkbox"/>	PC	Windows	4	Agora	0	1	
<input type="checkbox"/>	PC	Windows	3	Agora	0	1	
<input type="checkbox"/>	PC	Windows	2	Agora	0	1	
<input type="checkbox"/>	PC	Windows	1	Agora	0	1	

6 entitys

FILTER

By name

All

Windows





ADD ENTITY +

This is the [Admin panel](#) for [Entities](#). Here you can see the information on the different [Entities](#) you've created.

When pressing the [Add Entity button](#) you will be greeted by this screen:

Home › [lisy_Landing](#) › [Entities](#) › Add entity

Add entity

EntityType:	<div><div></div><div></div></div>  
Email:	<input type="text"/>
Name:	<input type="text"/>
Department:	<div><div></div><div></div></div>  
Location:	<input type="text" value="1"/>
Quantity:	<input type="text" value="1"/>
HoursBetweenTickets:	<input type="text" value="1"/>
<input type="checkbox"/> ShouldHaveTextBox	
<input type="checkbox"/> ShouldHaveInfoBox	
InfoBoxContent:	<input type="text"/>
ID:	-
Uuid:	1
Slug:	-
NumberOfOngoingTickets:	0

Save and add another

Save and continue editing

SAVE

Step by step:

1. First step is to select which [Entity Type](#) this [Entity](#) will belong to. This field is not mandatory, but it is recommended.
2. Second, we have the [Email field](#), this [Entity](#) will use the [Entity Type's Email address](#) if you leave it blank or you can input one of your own to override it. This field is not mandatory, but it is recommended.
3. Third, we have the [Name](#) of the [Entity](#). This field is mandatory.
4. Fourth, we have the [Department](#) field. Here is where you select under which [Department](#) this [Entity](#) should be stored. This field is mandatory.

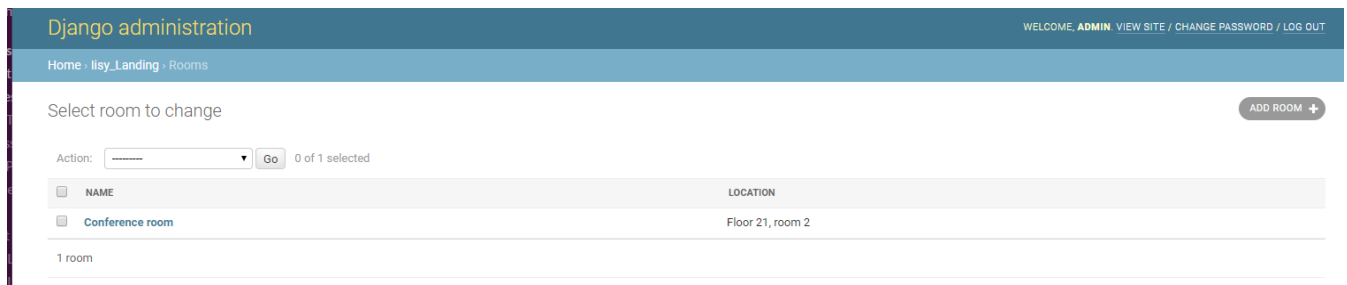
5. Fifth, we have the [Location](#) of the [Entity](#). We recommend you leave it as default until you have placed the [QR code](#). After that you can scan it and edit the [Location](#) on the spot. This field is mandatory.
6. Sixth, we have the [Quantity](#) field. This is where you enter how many [Entities](#) with this exact information you want to create. This field is mandatory.
7. Seventh, we have the [HoursBetweenTickets](#) field. This is where you enter the hourly interval that is needed before a new [Ticket](#) can be created. Do not input anything less than 0. This field is mandatory
8. Eighth, we have the [ShouldHaveTextBox](#) field. By marking the box you select whether you want the [Report page](#) to display a text box where you can input more information. This field is not mandatory.
9. Ninth, we have the [ShouldHaveInfoBox](#) field. By marking the box you select whether you want the [Report page](#) to display a link with information pertaining the [Entity](#), this could be used to give common troubleshooting tips. This field is not mandatory.
10. Tenth, we have the [InfoBoxContent](#) field. This is the information that will be displayed if you select the [ShouldHaveInfoBox](#) field. This field is not mandatory.
11. Eleventh, we have the [ID](#) field. This is the number created when the [Entity](#) was created, this is only used for identifying which [QR code](#) belongs to which [Entity](#). This field is created automatically and can't be changed.
12. Twelfth, we have the [Uuid](#) field. This is the randomized field that is used to create the [Slug](#) field. This field is created automatically and can't be changed.

13. Thirteenth, we have the [Slug](#) field. This is the [URL](#) that is created for the [Entity](#). This [URL](#) is used to create the [QR code](#) linking to the [Entity's Reporting page](#). This field is created automatically and can't be changed.
14. Fourteenth, we have the [NumberOfOngoingTickets](#) field. This is the field reporting the number of active [Tickets](#) linked to the Entity. This field is created automatically and changes automatically.

You can edit a desired [Entity](#) by clicking on its [EntityType](#) in the [Admin panel](#) . You can also delete it by clicking its box and choosing [Delete selected Entity](#) in the [Action list](#). To create the [QR code](#) for an [Entity](#) you click its box and choose [Make qr codes for selected devices](#) in the [Action list](#). It is possible to select multiple at once.

Room

Here is where you create your desired [Room](#). Think of a [Room](#) as an [Entity](#) filled with multiple [Entity Types](#). You can use a [Room](#) instead of an [Entity](#) if you have a room filled with multiple different devices but you don't want to create an [Entity](#) for every single one.



This is the [Admin panel](#) for [Rooms](#). Here you can see the information on the different [Rooms](#) you've created.

When pressing the [Add Room button](#) you will be greeted by this screen:

Home › [Iisy_Landing](#) › [Rooms](#) › Add room

Add room

Name:

Location:

Types:

Audio

PC

+

Hold down "Control", or "Command" on a Mac, to select more than one.

HoursBetweenTickets:

Department:

✎

+

Uuid:

1

Slug:

-

Save and add another

Save and continue editing

SAVE

This works much like when creating an [Entity](#).

1. First you enter the [Name](#) of the room. This field is mandatory.
2. Second, we have the [Location](#) of the [Room](#). We recommend you leave it as default until you have placed the [QR code](#). After that you can scan it and edit the [Location](#) on the spot. This field is mandatory.
3. Third, we have the [Types](#) field. This is where you select which [Entity types](#) exist inside this [Room](#). This field is mandatory.
4. Fourth, we have the [HoursBetweenTickets](#) field. This is where you enter the hourly interval that is needed before a new [Ticket](#) can be created. Do not input anything less than 0. This field is mandatory.
5. Fifth, we have the [Department](#) field. Here is where you select under which [Department](#) this [Entity](#) should be stored. This field is mandatory.

6. Sixth, we have the [Uuid](#) field. This is the randomized field that is used to create the [Slug](#) field. This field is created automatically and can't be changed.
7. Eighth, we have the [Slug](#) field. This is the [URL](#) that is created for the [Entity](#). This [URL](#) is used to create the [QR code](#) linking to the [Entity's Reporting page](#). This field is created automatically and can't be changed.

You can edit a desired [Room](#) by clicking on its [Name](#) in the [Admin panel](#). You can also delete it by clicking its box and choosing [Delete selected Room](#) in the [Action list](#). To create the [QR code](#) for an [Room](#) you click its box and choose [Make qr codes for selected rooms](#) in the [Action list](#). It is possible to select multiple at once.

Tickets

Here is where the [Tickets](#) for the [Entities](#) and [Rooms](#) are stored. They are created automatically by your users. Their information will be stored here and if you entered an [Email](#) for the [Room](#) or [Entity](#) you will also receive the information it as an email.

Home > [Iisy_Landing](#) > Tickets

Select ticket to change

Action: Go

0 of 1 selected

<input type="checkbox"/>	ENTITY	ROOM	DEPARTMENT	TYPE	MESSAGE	LOCATION	CREATED	STATUS
<input type="checkbox"/>	-	Conference room	Agora	Audio	Issue received for an object of type: Audio in the: Conference room in Floor 21, room 2	Floor 21, room 2 QR code: 1	March 10, 2020, 4:32 p.m.	Unresolved

1 ticket

Save

FILTER

By created

Any date
Today
Past 7 days
This month
This year
No date
Has date

By entity

All
Windows
Windows
Windows
Windows
Windows
Windows
Windows

This is the [Admin panel](#) for [Tickets](#). Here you can see the information of the different [Tickets](#) created. You can click the [Entity](#) of the [Ticket](#) and you will see more of its information.

Change ticket

HISTORY

Department:

Agora

Entity:

Room:

Conference room

Type:

Audio

☒ IsRoom

Status:

Unresolved

Message:

Issue received for an object of type: Audio in 1

Location:

Floor 21, room 2 QR code: 1

Delete

Save and add another

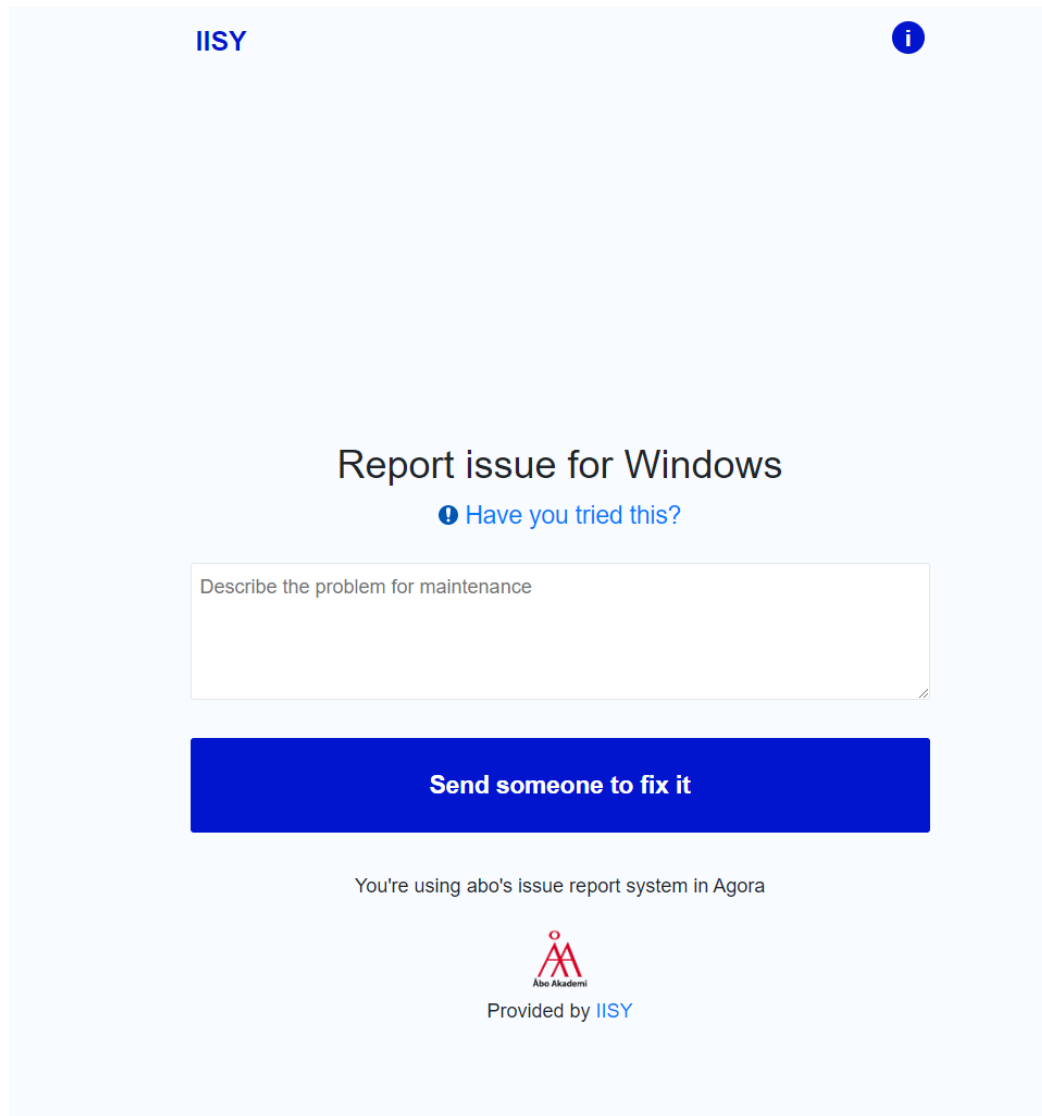
Save and continue editing

SAVE

You can edit the [Ticket](#) but changing anything but the [Status](#) isn't recommended.

Report page

The [Report page](#) is what your users interact with. It is automatically created for the [Entity](#) or [Room](#) you create and the [URL](#) is the [Slug](#) field. When a [User](#) scans a [QR code](#) a [Reporting page](#) like this is opened:



The screenshot shows a web interface for reporting an issue. At the top left is the 'IISY' logo, and at the top right is an information icon (a blue circle with a white 'i'). The main heading is 'Report issue for Windows'. Below this is a link with an exclamation mark icon that says 'Have you tried this?'. There is a large text input box with the placeholder text 'Describe the problem for maintenance'. Below the input box is a prominent blue button with the text 'Send someone to fix it'. At the bottom, it says 'You're using abo's issue report system in Agora' followed by the 'Abu Ahseni' logo (a stylized red 'A' with a dot above it) and the text 'Provided by IISY'.

This is a [Reporting page](#) that is created when you create an [Entity](#) with [ShouldHaveTextBox](#) and [ShouldHaveInfoBox](#) enabled. What the [User](#) enters in the [Text box](#) is what you will see in the [Message](#) field for the [Ticket](#), you will also receive the message in an email if you entered one.

IISY



Provided by IISY

Report issue for Audio

Send someone to fix it

Report issue for PC

Send someone to fix it

You're using abo's issue report system in Agora



This is a [Reporting page](#) for a [Room](#) that was created with an [Audio type](#) and a [PC type](#). If the [Audio button](#) is clicked then the [Ticket](#) will use the [Email](#) from [Audio type](#) to send an email with a [message](#).