
uniTicket

Release 0.1

Università della Calabria

17 feb 2023

1	Requisiti	3
2	Download del software e dipendenze	5
3	Setup parametri	7
4	Creazione Database	15
5	Creazione tabelle e superuser	17
6	Template Bootstrap Italia	19
7	Run	21
8	Produzione	23
9	Migrazione dalla v1.x alla v2.0.0	25
10	Ticket	29
10.1	Stati del ticket	31
10.2	Priorità	31
11	Tipologie di richieste	33
11.1	Stati	33
11.2	Livelli di visibilità in fase di creazione ticket	33
11.3	Ticket accessibili a utenti anonimi	34
11.4	Tipo di richiesta “Notifica“	34
11.5	Moduli di input	34
11.6	Clausole di accettazione obbligatorie	34
11.7	Attività predefinite	34
12	Attività	35
13	Strutture	37
14	Uffici	39
14.1	Stati	39
14.2	Ufficio predefinito della struttura	39

15	Manager	41
16	Operatore	43
17	Utente	45
18	Ticket	47
18.1	Presa in carico e impostazione priorità	47
18.2	Dettaglio ticket	48
18.3	Aggiornamento priorità	49
18.4	Gestione competenza uffici	49
18.5	Attività del ticket	50
18.6	Dipendenze da altri ticket	51
19	Chat	53
20	Videoconferenza	55
21	Dashboard	57
22	Tipologie di richieste	59
22.1	Aggiungi nuova	59
22.2	Dettaglio, modifica e competenza ufficio	60
22.3	Moduli di input	61
22.4	Clausole di accettazione	62
23	Uffici	65
23.1	Aggiungi nuovo	65
23.2	Dettaglio, modifica e assegnazione operatori	65
24	Dashboard	67
25	Dashboard	69
26	Ticket	71
26.1	Apertura nuovo ticket	71
26.2	Modifica ticket	71
26.3	Chiusura ticket	72
26.4	Eliminazione ticket non preso in carico	72
27	Messaggi	73
28	Configurazione	75
29	Template Bootstrap Italia (Linee Guida di Design pe la P.A.)	77
30	Gestione allegati firmati digitalmente	79
31	Costruzione dinamica dei form	81
32	Gestione Javascript di Django Formsets	83
33	JQuery Datatables	85
34	Single Sign On (SAML2)	87

Django **Unified Ticket System (uniTicket)** è un sistema software di ticketing ed helpdesk per il tracciamento e la digitalizzazione dei flussi di richieste e documentali.

Principali caratteristiche:

- Multi Tenant. Un unico sistema a supporto di diversi uffici e aree organizzative;
- Possibilità di trasferire e condividere ticket tra differenti uffici/aree;
- Possibilità di aggiungere clausole di consenso da accettare prima di aprire un ticket;
- Interdipendenza tra ticket;
- Lista di attività atomiche per ogni ticket, per guidare l'utente nel processo di lavorazione ed evasione;
- Form builder, possibilità di creare i moduli di inserimento per ogni categoria di ticket;
- Campi di input personalizzati, campi complessi e Django Formsets configurabili tramite widget;
- Gestione allegati firmati digitalmente (PDF e P7M), con controllo e validazione dell'integrità dei dati;
- Chat e videoconferenza per operatori e utenti;
- Report di riepilogo via email che include la lista dei ticket pendenti agli operatori;
- JQuery Datatables integrato, per una gestione Ajax con processamento lato server dei dati, per ottime performance;
- Template grafico Bootstrap Italia (Linee Guida di Design pe la P.A.) responsive per una esperienza di navigazione ottimale anche sui dispositivi mobili;
- Integrazione SAML2 SSO (pySAML2);

Github: <https://github.com/UniversitaDellaCalabria/uniTicket>

CAPITOLO 1

Requisiti

```
sudo apt install xmlsec1 mariadb-server libmariadbclient-dev python3-dev python3-pip  
↳ libssl-dev libmariadb-dev-compat libsasl2-dev libldap2-dev  
  
pip3 install virtualenv  
virtualenv -ppython3 helpdesk.env  
source helpdesk.env/bin/activate
```

Download del software e dipendenze

```
git clone https://github.com/UniversitaDellaCalabria/uniTicket.git
cd uniTicket
pip3 install -r requirements
```

Convert HTML to PDF using Webkit (QtWebKit)

```
wget https://github.com/wkhtmltopdf/wkhtmltopdf/releases/download/0.12.5/wkhtmltox_0.12.5-1.buster_amd64.deb
sudo dpkg -i wkhtmltox_0.12.5-1.buster_amd64.deb
sudo apt install -f
```


Setup parametri

```
cd uni_ticket_project

# copy and modify as your needs
cp settingslocal.py.example settingslocal.py
```

Nel file di configurazione generale **uni_ticket_project/settingslocal.py** è possibile:

- Aggiungere/disabilitare applicazioni django in *INSTALLED_APPS*
- Definire il model da utilizzare per la gestione degli utenti

```
# user model fpr auth
AUTH_USER_MODEL = "accounts.User"
```

- Definire i formati delle date da utilizzare

```
DEFAULT_DATE_FORMAT = '%Y-%m-%d'
DEFAULT_DATETIME_FORMAT = '{} %H:%M'.format(DEFAULT_DATE_FORMAT)
DATE_INPUT_FORMATS = [DEFAULT_DATE_FORMAT, '%d/%m/%Y']

# override globals
DATE_INPUT_FORMATS = [DEFAULT_DATE_FORMAT, '%Y-%m-%d']
DATETIME_INPUT_FORMATS = [DEFAULT_DATETIME_FORMAT, f'%Y-%m-%d {DEFAULT_TIME_FORMAT}']

# for javascript datepickers
# BootstrapItalia datepicker
JS_DEFAULT_DATE_FORMAT = "dd/MM/yyyy"
# Cutstom datettimepicker
JS_DEFAULT_DATETIME_FORMAT = 'DD/MM/Y hh:mm'
```

- Definire l'ADMIN_PATH
- Definire i database

```
DATABASES = {
  'default': {
    'ENGINE': 'django.db.backends.mysql',
    'NAME': 'uniauth',
    'HOST': 'localhost',
    'USER': 'that-user',
    'PASSWORD': 'that-password',
    'PORT': '',
    'OPTIONS': {'init_command': "SET sql_mode='STRICT_TRANS_TABLES'"}
  },
}
```

- Selezionare i widget da applicare ai campi dei form

```
CUSTOM_WIDGETS = {
  'BaseDateField': 'bootstrap_italia_template.widgets.BootstrapItaliaDateWidget',
  # 'BaseDateTimeField': 'bootstrap_italia_template.widgets.BootstrapItaliaTimeWidget',
  # 'CustomSelectBoxField': 'bootstrap_italia_template.widgets.BootstrapItaliaSelectWidget',
  ↵,
  'CustomRadioBoxField': 'bootstrap_italia_template.widgets.BootstrapItaliaRadioWidget',
  ↵,
  # 'BaseDateField': 'django.forms.widgets.DateInput',
  # 'DateField': 'django.forms.widgets.DateInput',
  # 'CustomSelectBoxField': 'django.forms.widgets.Select',
  # 'CustomRadioBoxField': 'django.forms.widgets.RadioSelect',
}
```

- Definire parametri relativi alla configurazione delle app «chat» e «channels»

```
# chat: message to load in a conversation from history
MESSAGES_TO_LOAD = 1500

if "channels" in INSTALLED_APPS:
    ASGI_APPLICATION = 'uni_ticket_project.routing.application'
    CHANNEL_LAYERS = {
        'default': {
            'BACKEND': 'channels_redis.core.RedisChannelLayer',
            'CONFIG': {
                "hosts": [('127.0.0.1', 6379)],
            },
        },
    },
}
```

- Definire i parametri per la criptazione basata su RSA dei token che viaggiano negli URL

```
# UNITICKET JWE support
UNITICKET_JWE_RSA_KEY_PATH = 'saml2_sp/saml2_config/certificates/key.pem'
UNITICKET_JWE_ALG = "RSA-OAEP"
UNITICKET_JWE_ENC = "A128CBC-HS256"
# end JWE support
```

- Definire `secret_key` e `salt` per la criptazione del codice CAPTCHA

```
# CAPTCHA encryption
CAPTCHA_SECRET = b'secret'
CAPTCHA_SALT = b'salt'
# end CAPTCHA encryption
```

- Definire la validità del CAPTCHA (in millisecondi)

```
CAPTCHA_EXPIRATION_TIME = 45000 # milliseconds
```

- Configurare le impostazioni del protocollo informatico (ArchiPRO)

```
# PROTOCOLLO, questi valori possono variare sulla base di come
# vengono istruite le pratiche all'interno del sistema di protocollo di riferimento

CLASSE_PROTOCOLLO = 'archipro_ws.protocollo'

# XML flusso
PROTOCOL_XML = """<Segnatura xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
↳xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<Intestazione>
<Oggetto>{oggetto}</Oggetto>
<Identificatore>
<CodiceAmministrazione>UNICAL</CodiceAmministrazione>
<CodiceA00>{a00}</CodiceA00>
<Flusso>E</Flusso>
</Identificatore>
<Mittente>

<Dipendente id="{identificativo_dipendente}">
<Denominazione>{denominazione_persona}</Denominazione>
</Dipendente>

<Studente id="{identificativo_utente}">
<Denominazione>{denominazione_persona}</Denominazione>
</Studente>

<Persona id="{id_persona}">
<Nome>{nome_persona}</Nome>
<Cognome>{cognome_persona}</Cognome>
<Denominazione>{denominazione_persona}</Denominazione>
</Persona>

</Mittente>
<Destinatario>
<Amministrazione>
<Denominazione>UNICAL</Denominazione>
<CodiceAmministrazione>UNICAL</CodiceAmministrazione>
<IndirizzoTelematico tipo="smtp">amministrazione@pec.unical.it</IndirizzoTelematico>
<UnitaOrganizzativa id=""/>
</Amministrazione>
</Destinatario>
<Classifica>
<CodiceTitolario>{id_titolario}</CodiceTitolario>
```

(continues on next page)

```

</Classifica>
<!-- Informazioni sul fascicolo -->
<Fascicolo numero="{fascicolo_numero}" anno="{fascicolo_anno}"/>
</Intestazione>
<Descrizione>
<Documento id="1" nome="{nome_doc}">
<DescrizioneDocumento>{nome_doc}</DescrizioneDocumento>
<TipoDocumento>{tipo_doc}</TipoDocumento>
</Documento>
<Allegati>
<!-- Allegati -->
</Allegati>
</Descrizione>
<ApplicativoProtocollo nome="ArchiPRO">
<Parametro nome="agd" valore="{agd}"/>
<Parametro nome="uo" valore="{uo}"/>
</ApplicativoProtocollo>
</Segnatura>
""""

# DEFAULT EMAIL
PROT_EMAIL_DEFAULT = 'default@email.com'

# TEST
PROT_TEST_AOO = 'default_aoo'
PROT_FASCICOLO_DEFAULT = 'default_fascicolo'
PROT_FASCICOLO_ANNO_DEFAULT = 'default_year'
PROT_AGD_DEFAULT = 'default_agd'
PROT_UO_DEFAULT = 'default_uo'
# PROT_UO_ID_DEFAULT = 'default_uo_id'
PROT_TITOLARIO_DEFAULT = 'default_titolario'

PROT_URL = 'url_test'
PROT_TEST_LOGIN = 'test_login'
PROT_TEST_PASSW = 'test_passw'

```

- Consentire ai super utenti Django di accedere a tutte le strutture in frontend

```

# superusers view all
SUPER_USER_VIEW_ALL = True

```

- Definire i parametri per la localizzazione

```

# Internationalization
# Set to False to avoid problems with javascript datepickers
# (that use the DATE_INPUT_FORMATS and DATETIME_INPUT_FORMATS)
# The template uses {% localize on %} tag to localize dates
USE_L10N = False

# localization
LANGUAGES = (
    ('it', _('Italiano')),

```

(continues on next page)

(continua dalla pagina precedente)

```
( 'en', _('Inglese')),
)

LANGUAGE_CODE = 'it'
LOCALE_PATHS = (
    os.path.join(BASE_DIR, "locale"),
)

TIME_ZONE = 'Europe/Rome'
```

Nel file di configurazione `uni_ticket/settings.py` è possibile individuare (ed eventualmente sovrascrivere in `settingslocal.py`):

- I nomi delle cartelle nelle quali verranno conservati gli allegati

```
# system attachments folders
LOGOS_FOLDER = 'logos'
STRUCTURES_FOLDER = 'structures'
TICKET_ATTACHMENT_FOLDER = 'ticket'
TICKET_CATEGORIES_FOLDER = 'categories'
TICKET_MESSAGES_ATTACHMENT_SUBFOLDER = 'messages'
TICKET_TASK_ATTACHMENT_SUBFOLDER = 'task'
CATEGORY_CONDITIONS_ATTACHMENT_SUBFOLDER = 'conditions'
```

- Il parametro che consente di mostrare la priorità dei ticket agli utenti

```
# show ticket priority to simple userse
SIMPLE_USER_SHOW_PRIORITY = False
```

- ID e Label del checkbox di accettazione delle clausole obbligatorie

```
# category conditions form field
TICKET_CONDITIONS_FIELD_ID = 'condizioni_field_id'
TICKET_CONDITIONS_TEXT = _('Dichiara altresì di aver letto '
                            'e compreso quanto scritto sopra '
                            'e di assumere ogni responsabilità '
                            'su quanto di seguito dichiarato')
```

- La denominazione dei campi *oggetto* e *descrizione* dei form per la creazione dei ticket

```
# new ticket heading text (user informations)
SHOW_HEADING_TEXT = True
TICKET_HEADING_TEXT = _('Soggetto richiedente: <b>{user}</b>'
                        '<br><span class="x-small">[taxpayer]</span>')

# new ticket static form fields
# ticket subject
TICKET_SUBJECT_ID = 'ticket_subject'
TICKET_SUBJECT_LABEL = _('Oggetto della Richiesta')
TICKET_SUBJECT_HELP_TEXT = _("Ulteriore specificazione o "
                             "personalizzazione dell'Oggetto della Richiesta")

# ticket description
```

(continues on next page)

(continua dalla pagina precedente)

```
TICKET_DESCRIPTION_ID = 'ticket_description'  
TICKET_DESCRIPTION_LABEL = _('Descrizione')  
TICKET_DESCRIPTION_HELP_TEXT = ('Ulteriore Descrizione della Richiesta, '  
                                'eventuali note del Richiedente')
```

- I livelli di priorità da assegnare ai ticket

```
PRIORITY_LEVELS = (  
    ('-2',_('Molto alta')),  
    ('-1',_('Alta')),  
    ('0',_('Normale')),  
    ('1',_('Bassa')),  
    ('2',_('Molto bassa')),  
)
```

- La soglia massima di ticket giornalieri per utente

```
# 0 = unlimited  
MAX_DAILY_TICKET_PER_USER = 10
```

- La denominazione di ogni tipologia di utente per la definizione degli URL

```
# user contexts  
CONTEXT_SIMPLE_USER = _('Utente')  
  
# To change the URLs prefix for every user type  
MANAGER_PREFIX = 'Manager'  
OPERATOR_PREFIX = 'Operatore'  
USER_PREFIX = 'user'  
  
# Do not edit! - START  
MANAGEMENT_URL_PREFIX = {'manager': MANAGER_PREFIX,  
                          'operator': OPERATOR_PREFIX,  
                          'user': USER_PREFIX}  
# Do not edit! - END
```

- Le definizioni per competenza abbandonata/sola lettura

```
# ticket competence abandoned  
NO_MORE_COMPETENCE_OVER_TICKET = _("Nessuna competenza sul ticket")  
# ticket readonly access  
READONLY_COMPETENCE_OVER_TICKET = _("Hai accesso al ticket in sola lettura")
```

- Il numero minimo di digits per la compressione del contenuto di un ticket

```
# min ticket content length (digits) to compress  
TICKET_MIN_DIGITS_TO_COMPRESS = 90
```

- La definizione degli utenti «employee» e «internal user» in base al tipo di organizzazione (università o altro)

```
# This parameters define the roles of users to open ticket  
# If True, an employee is a user that has this parameter filled (in user model)  
# If False, an employee is a user that is mapped as OrganizationalStructureOfficeEmployee
```

(continues on next page)

(continua dalla pagina precedente)

```
EMPLOYEE_ATTRIBUTE_NAME = 'identificativo_dipendente'  
EMPLOYEE_ATTRIBUTE_LABEL = 'Matricola dipendente'  
# Label  
ORGANIZATION_EMPLOYEE_LABEL = 'Dipendenti'  
# If True, an internal user (not guest) is a user that has this filled (in user model)  
# If False, an internal user is a user that is mapped as_  
↳OrganizationalStructureOfficeEmployee  
USER_ATTRIBUTE_NAME = 'identificativo_utente'  
USER_ATTRIBUTE_LABEL = 'Matricola studente'  
# Label  
ORGANIZATION_USER_LABEL = 'Studenti'
```

- I testi delle email che il sistema invia agli utenti
- Disabilitare la modalità *DEBUG* per la messa in produzione (Attenzione: il servizio in produzione richiede HTTPS)

Creazione Database

```
# create your MysqlDB
export USER='that-user'
export PASS='that-password'
export HOST='%'
export DB='uniauth'

# tested on Debian 10
sudo mysql -u root -e "\
CREATE USER IF NOT EXISTS '${USER}'@'${HOST}' IDENTIFIED BY '${PASS}';\
CREATE DATABASE IF NOT EXISTS ${DB} CHARACTER SET = 'utf8' COLLATE = 'utf8_general_ci';\
GRANT ALL PRIVILEGES ON ${DB}.* TO '${USER}'@'${HOST}';"
```

Creazione tabelle e superuser

```
./manage.py migrate  
./manage.py createsuperuser
```

Template Bootstrap Italia

Di default, il sistema si presenta con il template customizzato per l'Università della Calabria. Per utilizzare la versione standard di **Bootstrap Italia** basta modificare la riga 6 del file *uniTicket/uni_ticket_bootstrap_italia_template/base.html* come segue

```
{% extends 'bootstrap-italia-base.html' %}
```

e, se si desidera, commentare l'app *django_unical_bootstrap_italia* dalle `INSTALLED_APPS` in *settingslocal.py*.

CAPITOLO 7

Run

```
./manage.py runserver
```

Produzione

Ricorda di eseguire `compilemessages` per attuare la localizzazione e `compilescss/collectstatic` per compilare e copiare tutti i file statici nelle cartelle di produzione:

```
./manage.py compilemessages  
./manage.py compilescss  
./manage.py collectstatic
```

Per un ulteriore controllo in fase di debug è possibile utilizzare i comandi seguenti con uwsgi:

```
/etc/init.d/uni_ticket stop  
uwsgi --ini /opt/uni_ticket/uwsgi_setup/uwsgi.ini .debug
```

Migrazione dalla v1.x alla v2.0.0

Questa semplice guida consente di aggiornare agevolmente una istanza di uniTicket v1.x alla versione 2.

01 - Stappare il servizio

```
/etc/init.d/uni_ticket stop
```

02 - Export dei ContentType

E' necessario ricostruire la tabella dei ContentType, generata automaticamente da Django all'applicazione delle migrazioni, per mantenere la consistenza delle FK utilizzate dai log.

```
# CLI Django

from django.contrib.contenttypes.models import ContentType

ct = ContentType.objects.all()

# old_conf sarà quindi una lista di dizionari
old_conf = []
for cct in ct:
    old_conf.append({'pk': cct.pk,
                    'app_label': cct.app_label,
                    'model': cct.model})

print(old_conf)
```

Copiare old_conf su un file di testo, ci servirà dopo

02 - DB Backup

```
# CLI Django

./manage.py dumpdata --exclude auth.permission --exclude contenttypes --exclude sessions_
↪--indent 2 > path_to_your_file.json
```

03 - Path del progetto

- Rinominare la folder del progetto [path]/uniticket in [path]/uniticket_tmp
- Creare [path]/uniticket (è preferibile che si usi l'utente linux proprietario di [path]/uniticket)

04 - Download repository

```
# In [path]/uniticket
git clone https://github.com/UniversitaDellaCalabria/uniTicket.git
```

05 - Django settings

```
cp [path]/uniticket/uniticket/uni_ticket_project/settingslocal.py.example [path]/
↪uniticket/uniticket/uni_ticket_project/settingslocal.py
```

Modificare le variabili opportune (ignorare la parte dedicata al DB momentaneamente)

06 - Media e statics

Copiare *uniticket/data/media* e *uniticket/data/statics* in *uniticket/data*.

07 - Database

- Se si intende utilizzare lo stesso DB, eliminare tutte le tabelle presenti altrimenti utilizzare un nuovo DB (opzione consigliata)
- Aggiornare i dati relativi nel *settingslocal.py*

08 - Migrazioni

```
# CLI Django
./manage.py migrate
```

09 - Ripristino dei ContentType

Poichè i log dei ticket sono collegati ai ContentType, è necessario sovrascrivere i valori creati da Django nella migrazione iniziale per la consistenza del backup da importare.

Nel seguente script bisogna copiare il contenuto di *old_conf* stampato al punto **02 - Export dei ContentType**

```
# CLI Django

# in una variabile "old_conf" copiare la lista prodotta allo step 01
old_conf = ...

from django.contrib.contenttypes.models import ContentType

ct = ContentType.objects.all()

# cancella ContentType da aggiornare (quelli presenti nella lista)
to_delete = []
for cct in ct:
    app_label = cct.app_label
    model = cct.model
    for old_ct in old_conf:
        if old_ct['app_label'] == app_label and old_ct['model'] == model:
            to_delete.append(cct.pk)
```

(continues on next page)

(continua dalla pagina precedente)

```

        break
ct.filter(pk__in=to_delete).delete()

# aggiorna la pk dei ContentType rimasti
# per evitare che questa vada in conflitto con il successivo step di importazione
ct = ContentType.objects.all()
to_delete = []
for cct in ct:
    app_label = cct.app_label
    model = cct.model
    # nuova pk che non vada in conflitto con quelle da importare
    pk = cct.pk + 100
    # cambio il valore dei campi del contenttype di origine
    # con dei valori fake per permettere la creazione di uno nuovo
    cct.app_label = pk
    cct.model = pk
    cct.save()
    to_delete.append(cct.pk)
    # crea nuovo ContentType
    ContentType.objects.create(pk=pk, app_label=app_label, model=model)
ct.filter(pk__in=to_delete).delete()

# ripristina i contenttypes provenienti dal db di origine (lista old_conf)
for old_ct in old_conf:
    ContentType.objects.create(pk=old_ct['pk'],
                               app_label=old_ct['app_label'],
                               model=old_ct['model'])

```

10 - Load Data

Sostituire nel dump json le seguenti definizioni con nano (<https://it.stealthsettings.com/find-replace-nano-linux-os-x-terminal-text-editor.html>)

- matricola_dipendente => identificativo_dipendente
- matricola_studente => identificativo_utente

Se nel dump sono presenti le tabelle delle app chat e channels abilitarle nelle *INSTALLED_APPS* del settingslocal e applicare le eventuali migrazioni

```

# CLI Django

./manage.py loaddata path_to_your_file.json

```

11 - Campo <ticket.assigned_data>

Questo campo è presente e viene salvato automaticamente nella nuova release quando un ticket viene preso in carico la prima volta. Deve essere inizializzato per tutti i ticket con la data della prima presa in carico. Questo è necessario solo nel processo di migrazione dalla v1.x alla v2.x.

```

# CLI Django

from uni_ticket.models import Ticket, TicketAssignment

tickets = Ticket.objects.filter(assigned_date__isnull=True)

```

(continues on next page)

(continua dalla pagina precedente)

```
for ticket in tickets:
    first_taken = TicketAssignment.objects\
        .filter(ticket=ticket,
                taken_date__isnull=False)\
        .values_list("taken_date", flat=True)\
        .first()

    if first_taken:
        ticket.assigned_date = first_taken
        ticket.save()
        print("Assigned data update for ticket ", ticket.code)
```

12 - Se non ci sono criticità è possibile rimuovere la cartella [path]/uniticket_tmp

CAPITOLO 10

Ticket

Un ticket è una richiesta aperta, da un utente , per una specifica tipologia.

Francesco Filicetti (TEST) - Friday, 17 April 2020 Ruolo: Manager (Centro ICT d'Ateneo)

 **Nome Organizzazione**
Sottotitolo Organizzazione Seguici su   

[Dashboard](#) [Ticket](#) [Tipologie di richieste](#) [Uffici](#) [Messaggi](#) [Chat](#) [Configurazione](#) [Documentazione](#)

Ticket subject

5KWdtq75xUciBfBZcEwf8e

Attenzione
Il ticket non è stato ancora assegnato a nessun operatore negli uffici

- (Centro ICT d'Ateneo) Help-Desk**
 - + [clicca qui per prendere in carico il ticket](#)
 - + [clicca qui per assegnare a un operatore di questo ufficio](#)

Vedi questo messaggio in quanto **Manager** di **Centro ICT d'Ateneo**, **Operatore** dell'ufficio **Help-Desk** o Operatore degli uffici sopra elencati.

[← Torna ai ticket aperti](#)

Stato	Aperto
Priorità	Normale
Creato da	Francesco Filicetti (TEST) - Venerdì 17 Aprile 2020 12:40 E-mail: francesco.filicetti@unica1.it Codice fiscale: AAABBB11C22D333E Matricola dipendente: 999999 Matricola studente: 111111
Competenza	Competenza attiva (Centro ICT d'Ateneo) Help-Desk - Non ancora preso in carico in questo ufficio
Tipo di richiesta	Richiesta Software
Oggetto	Ticket subject
Descrizione	Ticket description

+ [Aggiornamenti di stato \(Log\) \(0\)](#)

 **Lorem Ipsum**
Lorem Ipsum sit amet

CONTATTI Comune di Lorem Ipsum Via Roma 0 - 00000 Lorem Ipsum Codice fiscale / P. IVA: 000000000 Posta Elettronica Certificata URP - Ufficio Relazioni con il Pubblico	LOREM IPSUM	SEGUICI SU     NEWSLETTER Form Newsletter
--	--------------------	--

[Media policy](#) [Note legali](#) [Privacy policy](#) [Mappa del sito](#) 

10.1 Stati del ticket

- **Aperto**

Appena creato, il ticket si trova in uno stato pendente.

Non ancora assegnato ad alcun operatore, esso può essere modificato, chiuso o eliminato dall'utente che lo ha creato.

- **Assegnato**

Una volta preso in carico dall'operatore competente, il ticket passa a questo stato e può essere gestito per l'evasione della richiesta.

L'utente che lo ha creato non può più apportare modifiche ma può chiuderlo in qualsiasi momento.

- **Chiuso**

Se il ticket è stato chiuso dall'utente prima di essere stato preso in carico da un operatore, esso non può essere riaperto. In caso contrario, un operatore con facoltà di gestione può riaprirlo e apportare ulteriori modifiche.

Un ticket può essere chiuso solo quando gli eventuali ticket da cui dipende e le eventuali attività collegate, tutti elementi vincolanti, sono stati chiusi.

10.2 Priorità

- **Molto alta**

- **Alta**

- **Normale** (priorità di default alla creazione del ticket)

- **Bassa**

- **Molto bassa**

Tipologie di richieste

Una tipologia di richiesta rappresenta l'ambito del ticket che viene aperto.
Ogni tipologia è associata a un ufficio, che ha la competenza della gestione delle richieste in arrivo.

11.1 Stati

- **Attiva** visibile agli utenti in fase di creazione del ticket
- **Non attiva** non visibile agli utenti

11.2 Livelli di visibilità in fase di creazione ticket

Ogni tipologia prevede 3 livelli di visibilità che consentono l'apertura di ticket a

- personale dell'organizzazione
- utenti dell'organizzazione
- ospiti
- utenti anonimi (la compilazione del form non richiede il login)

11.3 Ticket accessibili a utenti anonimi

Questa opzione consente agli utenti non loggati di accedere direttamente all'URL del form per l'apertura di un nuovo ticket. Il form, in questo caso, sarà provvisto di un codice CAPTCHA per evitare abusi. Una volta aperto, al ticket "anonimo" verrà assegnato, in modalità random e in qualità di proprietario (creatore), un operatore dell'ufficio di destinazione (se presente, altrimenti un manager della struttura o un operatore dell'ufficio speciale help-desk).

11.4 Tipo di richiesta "Notifica"

Se selezionata, questa opzione permetterà agli utenti di creare dei ticket che non seguono il normale flusso di gestione. Una richiesta di tipo notifica verrà chiuso automaticamente dal sistema, immediatamente dopo la sua apertura, e verrà assegnato, in modalità random, a uno degli operatori dell'ufficio competente (se assenti, a un manager/operatore help-desk della struttura). Casi d'uso: richieste di partecipazione ad eventi, semplici notifiche da parte degli utenti.

11.5 Moduli di input

Per essere attiva e visibile, una categoria deve prevedere un modulo di input attivo, che guiderà l'utente nell'apertura del ticket mediante compilazione dei campi ad esso collegati, e un ufficio collegato attivo, che avrà la competenza di gestirne le richieste.

11.6 Clausole di accettazione obbligatorie

Semplici condizioni che, se attive, l'utente è costretto ad accettare in fase di creazione di un ticket per la tipologia in questione.

11.7 Attività predefinite

Attività elementari che, se attive, saranno assegnate automaticamente a ogni ticket creato e ne condizioneranno la chiusura.

CAPITOLO 12

Attività

Un task elementare che ogni operatore può collegare a un ticket, in modo da definire una lista delle operazioni da compiere senza il completamento delle quali il ticket non può essere chiuso.

CAPITOLO 13

Strutture

Una struttura rappresenta una macro area (logica o fisica) dell'organizzazione presso cui il sistema viene utilizzato. Questa gestisce per competenza uno o più uffici secondo un principio di ripartizione delle responsabilità. In una Università, ad esempio, una struttura potrebbe essere rappresentata da un Dipartimento (o Facoltà).

Solo gli amministratori della piattaforma web (che accedono al backend di Django) possono creare/modificare/eliminare le strutture.

Micro-aree con una specifica sfera di competenza, afferenti a una struttura, a cui possono essere assegnati utenti operatori.

Ogni ufficio può gestire al più una categoria di ticket (o nessuna) e gli operatori ad esso assegnati saranno nelle condizioni di agire sui ticket appartenenti a quest'ultima.

14.1 Stati

- **Attivo:** visibile agli operatori in fase di gestione delle competenze
- **Non attivo:** non visibile agli operatori in fase di gestione delle competenze

14.2 Ufficio predefinito della struttura

A ogni struttura è collegato un ufficio “speciale”, di default “Help-Desk“, che dà la facoltà agli operatori ad esso afferenti di poter gestire tutti i ticket aperti in qualsiasi ufficio della struttura stessa.

Questo ufficio non può essere eliminato o reso inattivo.

Manager

Gestisce i ticket aperti verso qualsiasi ufficio della propria struttura.

Ha la facoltà di creare/modificare/eliminare categorie e uffici e definire l'afferenza degli operatori verso ognuno di essi.

Può creare nuovi moduli di input associati alle categorie e definire clausole di accettazione.

Per ogni ticket, può definire nuove assegnazioni, creare attività e aggiornare lo stato.

Un utente ha i privilegi di manager se afferisce all'*Ufficio predefinito della struttura* ed è presente nella lista di utenti in grado di gestire la struttura (*User Manage Organizational Structure*).

Solo gli amministratori della piattaforma web (che accedono al backend di Django) possono gestire queste liste di utenti e, quindi, creare nuovi manager.

CAPITOLO 16

Operatore

Gestisce il workflow dei ticket collegati agli uffici a cui afferisce.

Come il manager, può definire per questi nuove assegnazioni, creare attività e aggiornare lo stato.

Se assegnato all'*Ufficio predefinito della struttura* può gestire tutti i ticket della struttura.

Non ha alcuna facoltà sull'organizzazione di categorie e uffici.

CAPITOLO 17

Utente

Può aprire nuovi ticket e consultarne i dettagli.

Può modificare o eliminare un ticket prima che questo venga preso in carico e può chiuderlo in qualsiasi momento.

Durante la “vita” del ticket, può interagire con gli operatori mediante il pannello di messaggistica dedicato.

Non ha i permessi per l’interazione con le funzionalità gestionali.

La gestione del workflow di un ticket prevede strumenti utilizzabili sia dagli utenti **Manager** che dagli **Operatori**.

18.1 Presa in carico e impostazione priorità

La prima operazione da effettuare per la gestione di un ticket è la presa in carico, ovvero l'assunzione della responsabilità di gestione.

All'atto di questo aggiornamento di stato, all'utente operatore viene chiesto di scegliere il livello di priorità da assegnare al ticket, in modo da posizionarlo correttamente nella lista dei ticket aperti.

Attenzione

Il ticket non è stato ancora assegnato a nessun operatore negli uffici

- **(Centro ICT d'Ateneo) Accessi in VPN**
 - clicca qui per prendere in carico il ticket

Priorità: *

Normale

Prendi il ticket in carico

- clicca qui per assegnare a un operatore di questo ufficio

Priorità: *

Normale

Seleziona operatore: *

Assegna ticket a un operatore dell'ufficio

Vedi questo messaggio in quanto **Manager** di **Centro ICT d'Ateneo**, **Operatore** dell'ufficio **Help-Desk** o Operatore degli uffici sopra elencati.

Un utente manager può, in questa fase, assegnare il ticket a un operatore dell'ufficio competente.

18.2 Dettaglio ticket

La schermata di dettaglio fornisce tutte le informazioni necessarie, oltre a un set di strumenti di gestione. Oltre alla scheda dettagliata, è presente una lista con tutte le operazioni effettuate sul ticket, accompagnate dalla data e dall'utente responsabile. E' disponibile l'elenco delle attività collegate al ticket e quello dei ticket da cui esso dipende.

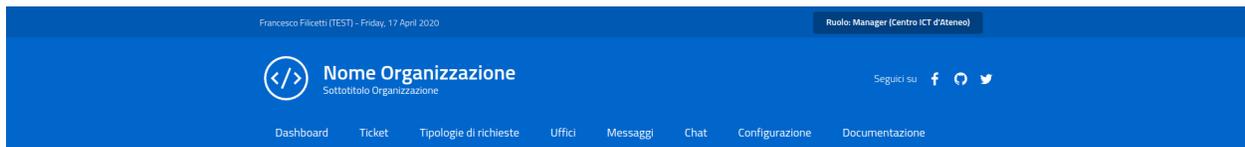
18.3 Aggiornamento priorità

In qualsiasi momento della vita del ticket è possibile aggiornare la sua priorità.

18.4 Gestione competenza uffici

E' possibile abbandonare o condividere la competenza del ticket con altri uffici, per motivi organizzativi e gestionali, o scegliere di trasferirla scegliendo di:

- abbandonare completamente i propri privilegi di gestione del ticket;
- mantenere ancora l'accesso ma in sola lettura.



Trasferisci competenza ticket

Ticket subject (5KWDtq75xUciBfBZcEwf8e)

[← Torna al ticket](#)

Struttura
 Centro ICT d'Ateneo

Categoria
 A chi è indirizzata la richiesta?

ⓘ Mantieni la possibilità di gestire il ticket:
 Deselezionando questa opzione non si avrà più accesso al ticket

Accedi al ticket in sola lettura:
 Selezionando questa opzione insieme a quella precedente si avrà accesso al ticket in sola lettura

Ufficio di origine
 Tutti quelli a cui afferisco in questa struttura

Ufficio a cui applicare i nuovi livelli di accesso (sola lettura, competenza abbandonata...)

[Trasferisci competenza](#)



18.5 Attività del ticket

L'aggiunta, la gestione e la cancellazione delle attività del ticket può essere effettuata con molta flessibilità finché esso si trova in stato «Assegnato».

Attività non chiuse impediscono la chiusura del ticket.

Oggetto: *

Testo: *

Priorità: *

Normale ▼

Allegato:

Scegli file Nessun file selezionato

✓ Crea attività

← Torna al ticket
✎ Modifica
✕ Elimina
✓ Chiudi

Stato	Aperto
Priorità	Normale
Creato da	Francesco Filicetti - Mercoledì 04 Settembre 2019 14:51
Ticket	Subject (o2oKH3Jc6hmje32EPnQQ62)
Oggetto	Attività 1
Descrizione	attività 1

+ Aggiornamenti di stato (0)

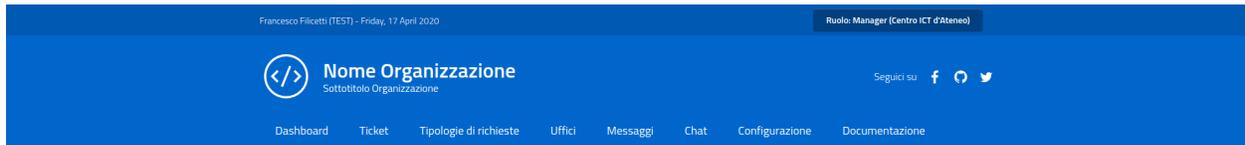
Priorità: *

Normale ▼

✓ Modifica la priorità

18.6 Dipendenze da altri ticket

Un operatore può assegnare o rimuovere una dipendenza, scegliendo tra i ticket che sono sotto la sua responsabilità. Questo crea un legame forte, come per le attività, che impedisce al ticket in oggetto di essere chiuso finchè tutte le sue dipendenze attive non sono chiuse.



Aggiungi dipendenza ticket

Ticket subject (5KWDtq75xUciBfBZcEwf8e)

[← Torna al ticket](#)

Ticket: *

.....

Note: *

.....

[→ Assegna dipendenza](#)



- Dipendenze da altri ticket (1)

Subject 4 (FJ5fbEPaRxor65hwyT5yrd) **Aperto**

Notes

[Rimuovi dipendenza](#)

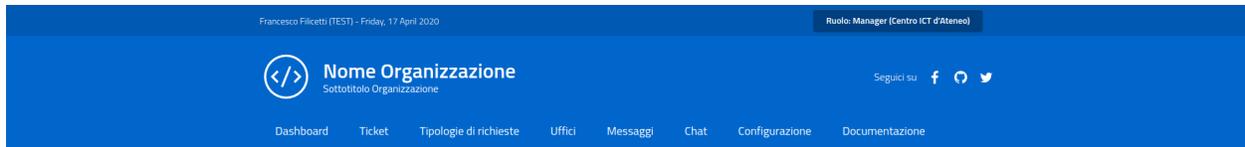
[+ Aggiungi nuova dipendenza](#)

CAPITOLO 19

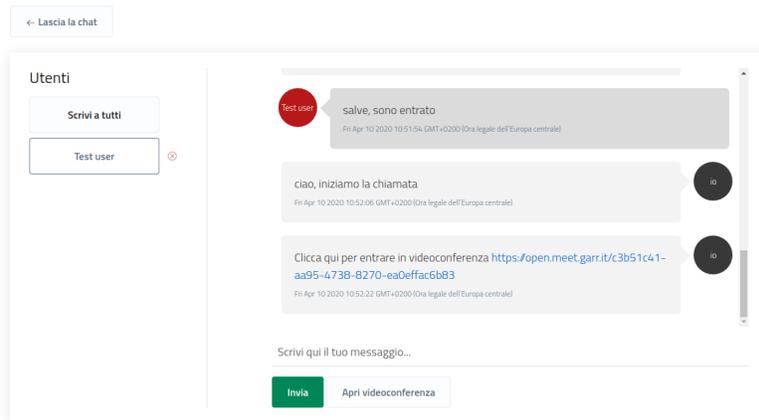
Chat

Per ogni struttura configurata all'interno del sistema, è concesso agli operatori di help-desk afferenti a questa di popolare una stanza di web chat dove accogliere gli utenti.

Gli utenti che accederanno alla chat vedranno la lista degli operatori disponibili. Gli operatori selezioneranno gli utenti con i quali stabilire la comunicazione. Ogni utente esporrà il proprio nome e cognome.



Centro ICT d'Ateneo [chat-room]



Questa funzionalità non vuole sostituirsi alla apertura dei Ticket ma piuttosto offrire l'opportunità agli utenti e agli operatori di confrontarsi direttamente ed eventualmente stabilire una *Videokonferenza*, dove chiarire e verificare in maniera più approfondita e diretta ogni aspetto oggetto di confronto.

CAPITOLO 20

Videoconferenza

Ogni operatore può concordare con uno o più utenti, mediante *Chat* o appuntamento concordato in risposta ad un Ticket, una sessione di video conferenza.

CAPITOLO 21

Dashboard

La dashboard per un utente Manager è un pannello dal quale si ha rapido accesso alla gestione di qualsiasi elemento del sistema (ticket, categorie, uffici).

A differenza degli altri utenti, infatti, il Manager ha accesso a tutte le sezioni.

Dashboard

Manager

6 Ticket in attesa 6 Ticket aperti 3 Ticket chiusi 1 Messaggi Categorie Ticket Uffici

Categorie Ticket

[Aggiungi nuova categoria](#)

Non attiva Categoria 1 descrizione categoria 1 Ufficio competente: Nessuno (competenza struttura) Ticket in questa categoria: 0 Clausele attive: 0	Non attiva Categoria 2 descrizione categoria 2 Ufficio competente: Nessuno (competenza struttura) Ticket in questa categoria: 0 Clausele attive: 0	Non attiva Categoria 3 descrizione categoria 3 Ufficio competente: Nessuno (competenza struttura) Ticket in questa categoria: 0 Clausele attive: 0
Attiva Categoria 4 descrizione categoria 4 Ufficio competente: Ufficio 3 Ticket in questa categoria: 0 Clausele attive: 0	Attiva Categoria 5 descrizione categoria 5 Ufficio competente: Ufficio 3 Ticket in questa categoria: 5 Clausele attive: 0	Attiva Categoria 6 descrizione categoria 6 Ufficio competente: Ufficio 1 Ticket in questa categoria: 1 Clausele attive: 1
Attiva Categoria 7 descrizione categoria 7 Ufficio competente: Ufficio 2 Ticket in questa categoria: 6 Clausele attive: 2	Attiva Categoria 8 descrizione categoria 8 Ufficio competente: Ufficio 2 Ticket in questa categoria: 1 Clausele attive: 0	

Uffici

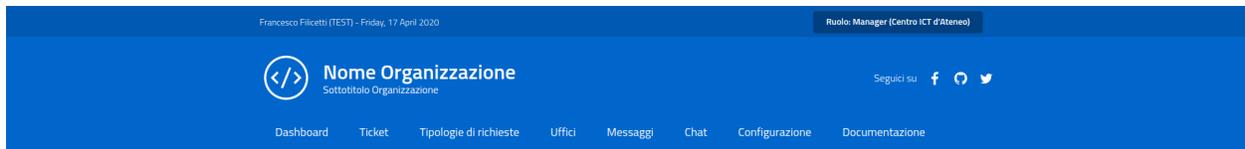
[Aggiungi nuovo ufficio](#)

Attivo Ufficio predefinito Help-Desk Help-Desk Office Operatori: 3 Ticket assegnati: 7 Categorie di competenza: Nessuna	Attivo Ufficio 1 descrizione ufficio 1 Operatori: 2 Ticket assegnati: 5 Categorie di competenza: Categoria 6	Non attivo Ufficio 2 descrizione ufficio 2 Operatori: 2 Ticket assegnati: 1 Categorie di competenza: Categoria 7
Attivo Ufficio 3 descrizione ufficio 3 Operatori: 2 Ticket assegnati: 0 Categorie di competenza: Categoria 4, Categoria 5	Non attivo Ufficio 4 descrizione ufficio 4 Operatori: 1 Ticket assegnati: 0 Categorie di competenza: Nessuna	Non attivo Ufficio 5 descrizione ufficio 5 Operatori: 1 Ticket assegnati: 0 Categorie di competenza: Nessuna
Non attivo Ufficio 6 descrizione ufficio 6 Operatori: 1 Ticket assegnati: 1 Categorie di competenza: Nessuna	Attivo Ufficio 7 descrizione ufficio 7 Operatori: 1 Ticket assegnati: 0 Categorie di competenza: Categoria 8	

22.1 Aggiungi nuova

La creazione di una nuova tipologia di richiesta prevede l'inserimento dei campi Nome e Descrizione, la scelta dei ruoli di accessibilità per l'apertura di nuove richieste, del tipo di richiesta e della possibilità di mostrare all'utente una intestazione con i suoi dati personali in fase di apertura di un nuovo ticket.

Una tipologia appena creata si presenta con stato *Non attiva*, perché sprovvista di un modulo di input attivo e di un ufficio attivo competente.



Nuova tipologia di richiesta

Crea una nuova tipologia di richieste nella struttura Centro ICT d'Ateneo

[← Dashboard](#)

Nome: *

Descrizione:

Messaggio di conferma:

Es: "Hai correttamente confermato la tua partecipazione". Apri e chiudi le parentesi graffe per inserire il codice del ticket. Lascia vuoto per usare il testo predefinito "Ticket "[]" creato con successo"

Messaggio se non attiva:

Viene mostrato agli utenti se cercano di accedere al form

Ticket di tipo Notifica:

Ticket che viene automaticamente preso in carico

Mostra agli utenti un testo di accettazione in fase di apertura nuovo ticket:

Accessibile a Utenti anonimi:

Accessibile a Ospiti:

Accessibile a Studenti:

Accessibile a Dipendenti:

[✓ Crea nuova tipologia di richiesta](#)



22.2 Dettaglio, modifica e competenza ufficio

La schermata di dettaglio, oltre alle informazioni della tipologia, presenta l'elenco dei moduli di input a essa associati e le relative clausole obbligatorie di accettazione.

Da questa sezione è possibile anche aggiornare l'ufficio competente, scegliendo dall'elenco di quelli attivi afferenti alla struttura.

← Dashboard
🏠 Torna alle categorie
✎ Modifica
☰ Aggiungi modulo
⌵ Aggiungi clausola
⊞ Disattiva
✖ Elimina

Denominazione	Categoria 4	
Stato	Attiva	
Descrizione	descrizione categoria 4	
Ufficio di competenza	(Struttura 1, Macro Area) Ufficio 3	Dettagli Rimuovi
Accessibile agli ospiti	×	
Accessibile agli utenti dell'organizzazione	×	
Accessibile agli operatori	✓	
+ Moduli di input (2)		
+ Clausole obbligatorie per i ticket (0)		

22.3 Moduli di input

Per essere visibile e attiva, una tipologia di richiesta deve avere, oltre a un ufficio competente, un modulo di input attivo.

- Moduli di input (2)

modulo 2	Sabato 10 Agosto 2019 12:48 non attivo	Vedi anteprima Dettagli Attiva modulo Elimina modulo
Modulo 1	Martedì 06 Agosto 2019 10:11 attivo	Vedi anteprima Dettagli Disattiva modulo Elimina modulo

Agli utenti manager è consentita la creazione di innumerevoli moduli per ogni tipologia, ma sempre uno sarà quello attivo.

Non è possibile eliminare moduli di input tramite i quali sono stati già creati dei ticket. Ogni ticket, infatti, è collegato a un preciso modulo e il suo workflow sarà operativo anche se il modulo dovesse essere disabilitato.

La costruzione di un modulo di input prevede la scelta e l'inserimento di singoli campi, selezionabili dall'elenco disponibile.

← Torna alla categoria ⌕ Vedi anteprima ✎ Rinomina ⌛ Disattiva ✖ Elimina

i Nessun campo di input personalizzato presente

Aggiungi un campo di input

Nome: *

Tipologia di campo: *

----- ▾

Lista di Valori:

Viene considerato solo se si sceglie 'Menu a tendina' oppure 'Serie di Opzioni'. (Es: valore1;valore2;valore3...)

Obbligatorio:

Aiuto:

Ordinamento:

0

posizione nell'ordinamento

+ Aggiungi campo di input

L'utente può visualizzare in qualsiasi momento l'anteprima del modulo, anche in fase di costruzione, così da rendersi conto di quello che la piattaforma presenterà all'utente all'atto della creazione di un nuovo ticket.

22.4 Clausole di accettazione

L'inserimento delle *Clausole di accettazione obbligatorie* è indipendente dal modulo utilizzato e si applica ai ticket creati per la tipologia in questione.

Una clausola può essere creata/modificata/disattivata/eliminata dal manager facilmente.

[← Torna alla categoria](#)

Titolo: *

Descrizione: *

Ordinamento:

0

Visibile nel documento di stampa:

Attiva:

[↻ Aggiungi clausola](#)

23.1 Aggiungi nuovo

L'aggiunta di un nuovo ufficio prevede l'inserimento dei campi Nome e Descrizione.
Un ufficio appena creato si presenta con stato "Non attivo".

23.2 Dettaglio, modifica e assegnazione operatori

La schermata di dettaglio, oltre alle informazioni, presenta l'elenco degli utenti afferenti alla struttura che potrebbero essere assegnati all'ufficio e quindi gestirne i ticket.
L'elenco degli operatori può essere aggiornato dagli utenti Manager in qualsiasi momento.

CAPITOLO 24

Dashboard

A differenza degli utenti Manager, un Operator non ha i privilegi per gestire Uffici e Categorie.
La sua Dashboard consente l'accesso alle liste di ticket e visualizza un riepilogo degli uffici a cui l'utente è assegnato.

mercoledì, 04 settembre 2019 Stai operando come: Struttura 2, Macro Area (Operatore)

Nome Organizzazione
Sottotitolo Organizzazione

Dashboard Ticket Messaggi Configurazione Documentazione

Dashboard

Operatore

1 Ticket in attesa 3 Ticket aperti 2 Ticket chiusi 0 Messaggi

Uffici di afferenza

Attivo Didattica Operatori: 1 Categorie di competenza: Categoria 9	Attivo Ufficio Tecnico Operatori: 1 Categorie di competenza: Dimes nuova
---	---

Lorem Ipsum
Lorem Ipsum sit amet

CONTATTI
Comune di Lorem Ipsum
Via Roma 0 - 00000 Lorem Ipsum Codice fiscale / P.
IVA: 0000000000
Posta Elettronica Certificata
URP - Ufficio Relazioni con il Pubblico

LOREM IPSUM

SEGUICI SU
It Twitter M Be

NEWSLETTER
Form Newsletter

Media policy Note legali Privacy policy Mappa del sito

Dashboard

Un utente utilizzatore nella dashboard visualizza il riepilogo di tutti i suoi ticket (in qualsiasi stato), ordinati per priorità decrescente.

mercoledì, 04 settembre 2019 Stai operando come: Utente semplice

Nome Organizzazione
Sottobito Organizzazione

Dashboard Apri nuovo ticket Messaggi Configurazione Documentazione

Seguici su [f](#) [t](#) [M](#) [Be](#)

Dashboard

Livello utente semplice

1 Ticket in attesa
1 Ticket aperti
0 Ticket chiusi
0 Messaggi
Apri un nuovo ticket

2019 Cerca...

Mostra
10

Codice	Oggetto	Categoria	Creato il	Priorità	Stato
o2oH3lcGImje32EPhPO6Z	Subject	Categoria 7	2019-09-04 13:25	Normale	In attesa di essere preso in carico
0XeMieZBauyWNjImEbk80	Problema umidità	Categoria 9	2019-09-04 13:25	Normale	Aperto

Visibili 2 di 2 record totali [Precedente](#) [Successiva](#)

Lorem Ipsum
Lorem ipsum sit amet

CONTATTI

Comune di Lorem Ipsum
Via Roma 0 - 00000 Lorem Ipsum. Codice fiscale / P. IVA: 000000000

Posta Elettronica Certificata
URP - Ufficio Relazioni con il Pubblico

LOREM IPSUM

SEGUICI SU

[f](#) [t](#) [M](#) [Be](#)

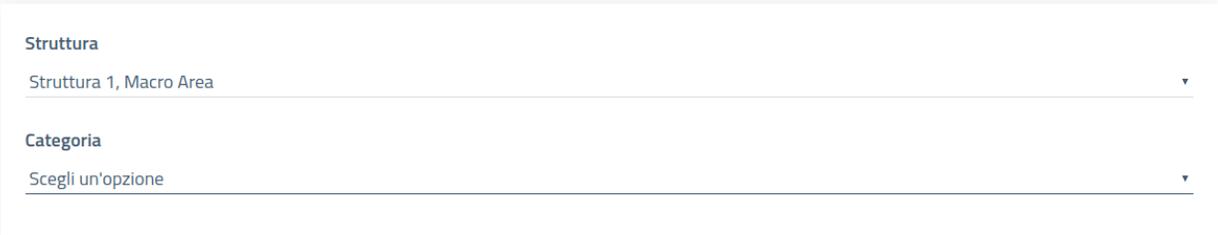
NEWSLETTER
Form Newsletter

Media policy Note legali Privacy policy Mappa del sito ↑

26.1 Apertura nuovo ticket

L'apertura di un nuovo ticket presuppone la scelta della struttura di destinazione e di una tipologia di richiesta. La tipologia deve essere attiva e accessibile dall'utente in base ai ruoli definiti.

Una volta scelti questi due parametri, all'utente viene presentato il modulo di input definito dal manager della struttura per quella particolare categoria.



The image shows a screenshot of a web form with two dropdown menus. The first dropdown is labeled 'Struttura' and has the text 'Struttura 1, Macro Area' selected. The second dropdown is labeled 'Categoria' and has the text 'Scegli un'opzione' selected. Both dropdowns have a small downward-pointing triangle on the right side.

26.2 Modifica ticket

Un utente utilizzatore può modificare un ticket esclusivamente se questo non è stato assegnato. La modifica può interessare qualsiasi dettaglio del ticket.

26.3 Chiusura ticket

In qualsiasi momento l'utente utilizzatore può chiudere uno dei suoi ticket (senza tener conto di eventuali dipendenze associate), a meno che questo non si trovi già in stato "Chiuso".

Questa funzionalità risulta essere utile nel caso in cui l'utente si dovesse rendersi conto che la sua richiesta è stata già soddisfatta diversamente o nel caso di richieste errate.

26.4 Eliminazione ticket non preso in carico

Come per la modifica, un utente può eliminare i suoi ticket se questi non sono stati ancora presi in carico.

CAPITOLO 27

Messaggi

Il pannello di messaggistica è disponibile per ogni ticket e consente lo scambio di messaggi tra utenti e operatori. Dal menu principale è possibile accedere alla voce “Messaggi” per il controllo di eventuali chat non lette.

Configurazione

Attivazione/Disattivazione (asincrona via Ajax) delle notifiche e-mail e consultazione dati personali.

[← Dashboard](#)

✓ Notifiche abilitate sulla mail: **francesco.filicetti@unical.it** ✕

✉ **Notifiche e-mail**

Se attiva, questa funzione invierà sulla mail **francesco.filicetti@unical.it** le notifiche inerenti ai tuoi ticket e un rapporto periodico di riepilogo. Disattivandola, non verrà recapitato alcun messaggio.

Nome	Manager
Cognome	Filicetti
E-mail	francesco.filicetti@unical.it
Codice fiscale	None

Template Bootstrap Italia (Linee Guida di Design pe la P.A.)

La libreria Bootstrap Italia è il modo più semplice e sicuro per costruire interfacce web moderne, inclusive e semplici da mantenere.

Completamente open-source, costruita sulle fondamenta di **Bootstrap 4.3.1**, di cui eredita tutte le funzionalità, componenti, griglie e classi di utilità, personalizzandole secondo le Linee Guida di Design per i siti web della Pubblica Amministrazione, Bootstrap Italia usa i pattern e i componenti definiti nello UI Kit di Designers Italia e li trasforma in codice già pronto all'uso!

<https://github.com/italia/design-django-theme>

Gestione allegati firmati digitalmente

Grazie all'integrazione della libreria Django Dynamic Form e di FileSig, la piattaforma è in grado di gestire file PDF firmati digitalmente e P7M e di effettuare una validazione delle firme, visualizzandone i dettagli.

Allegati



[pdf: pdf_firmato.pdf](#)

- Signature Validation: Signature is Valid.
- Signing Time: Oct 09 2017 17:43:26
- Signer full Distinguished Name:
dnQualifier=17414932,SN=SURNAME,givenName=SURNAME,serialNumber=IT:FISCALE_CODE,CN=FULL NAME,O=non presente,C=IT

Il Django Form Field utilizzato per questa feature è stato basato su: <https://github.com/peppelinux/FileSignatureValidator>

Costruzione dinamica dei form

App *django-form-builder* integrata per la costruzione di form dinamici e la definizione di nuovi campi custom.

<https://github.com/UniversitaDellaCalabria/django-form-builder>

Gestione Javascript di Django Formsets

Implementazione javascript del controllo frontend per la gestione di campi di input “complessi”, costruiti da field elementari e pertanto adattabili alle più disparate esigenze di configurazione dei moduli di input.

<https://docs.djangoproject.com/en/2.2/topics/forms/formsets/>

CAPITOLO 33

JQuery Datatables

Gestione ottimizzata dei record delle tabelle, che velocizza il caricamento delle pagine e non sovraccarica il server e il client in fase di renderizzazione dei risultati di una ricerca.

<https://datatables.net/>

<https://github.com/peppelinux/django-datatables-ajax>

Single Sign On (SAML2)

Federare uniTicket in un sistema SSO SAML2 è una operazione estremamente semplice. All'interno dei requirements già otteniamo le dipendenze a `djangosaml2` e `pysaml2`. Per federare uniTicket presso un IdP basterà ereditare la configurazione presso `saml2_sp/settings.py` e modificarla a proprio piacimento.

Nello specifico i parametri rilevanti sono:

- `entityid`;
- `required_attributes`;
- `metadata["remote"]`.

Modificare questi ultimi ed eventuali altri sulla base dei parametri tecnici per la federazione alla organizzazione di propria appartenenza. Per attivare la configurazione scelta basterà includere in `settingslocal.py` una dichiarazione di questo genere:

```
if 'saml2_sp' in INSTALLED_APPS:
    from saml2_sp.settings import *
```

Per rendere operativa l'autenticazione tramite SAML2 includere le app `djangosaml2` e `saml2_sp`

```
INSTALLED_APPS = [
    ...

    ##SAML2 SP
    'djangosaml2',
    'saml2_sp',
]
```

e commentare gli url login e logout nel file `urls.py` del progetto

```
# Login/Logou URLs
# for local auth
# disable these for SAML2 auth or others
```

(continues on next page)

(continua dalla pagina precedente)

```
path('{}/'.format(settings.LOGIN_URL), auth_views.LoginView.as_view(template_name='login.  
↪html'), name='login'),  
path('{}/'.format(settings.LOGOUT_URL), auth_views.LogoutView.as_view(template_name=  
↪'logout.html', next_page='../'), name='logout'),
```