Son Of X-Shooter

SOXS Scheduler design

Issue 1.3
Date: 2026-03-30

	NAME	SIGNATURE	DATE
PREPARED	R. Brivio F. Guidolin M. Landoni		2026-03-30
APPROVED	P. Schipani	P. Schipani	2026-03-30
RELEASED	S. Campana	Sergio Campana	2026-03-30

CHANGE RECORD

ISSUE	DATE	SECTION/PARAGRAPH AFFECTED	REASON/INITIATION DOCUMENTS/REMARKS
1.0	30/09/2023	All	First after-PDR version
1.1	01/10/2024	All	PAE delivered version
1.2	25/01/2025		Answer to RIX SOXS 243
1.3	30/03/2026	All	Updates of new functionalities and LaSilla reports in preparation of PAC

TABLE OF CONTENTS

1 Purpose and Scope of the Document

In this document, we provide the detailed software architecture, design and implementation of the SOXS scheduler.

2 List of Applicable and Reference Documents

2.1 Applicable Documents

Ref.	Document title	Document ID
AD1	Memorandum of Understanding for the Preliminary Design Phase of the SOXS Project *for the Preliminary Design Phase of the SOXS Project*	None
AD2	PDR operation plan	SOXS-PLA-0002 v1.0
AD3	Calibration plan	SOXS-PLA-0006 v1.0
AD4	SOXS MoU between ESO and INAF	None
AD5	ESO Data Interface Control Document (DICD)	ESO-044156
AD6	ESO Science Data Product standard	GEN-SPE-ESO-33000-5335
AD7	Scheduler interface User manual	SOXS-MAN-0007

2.2 Reference Documents

Ref.	Document title	Document ID
RD1	ESO Access data policy (web based)	None

3 Remote operations

3.1 Introduction and overview

SOXS is a single-object spectrograph with a resolution R~4,500 that is due to observe the sky from the ESO New Technology Telescope in La Silla. It will be operated in the ESO La Silla-Paranal Observatory operation environment, with no astronomer on the mountain.

From this stems the necessity to develop a new software with a high level of automation, that is able to make reliable decisions on its own and that can aid the operator during the night. The same software will also provide the Consortium with all the tools necessary to properly organize the night in advance.
Specifically, the minimum requirements for this scheduler are:

- Suggest a schedule for the night, respecting scientific priorities and maximizing the scientific output by optimizing observability constraints.
- Promptly provide on-the-fly backups for the night by convolving observable targets with current weather conditions (accessed through the ASM Weather API) and pointing restrictions.
- Balance and respect the Granted Time of Observation between the Consortium targets ToOs (being ingested via the Marshall Application) and regular ESO proposals.
- Keep adequate track of the operations of the instrument in terms of duty cycle, percentage of completed proposals, average observing conditions and hit/miss rate of transient events coming from the Consortia Targets.
- Communicate automatically with the official ESO Phase 2 tools to deliver the specifics of the OB (from the targets) using standard ESO templates.
- Autonomously populate the Visitor Execution Sequence (VES) according to the pre-approved or updated schedule for the night.

A sketch of the Scheduler Application, required for remotely operating SOXS, is reported in Figure 1.

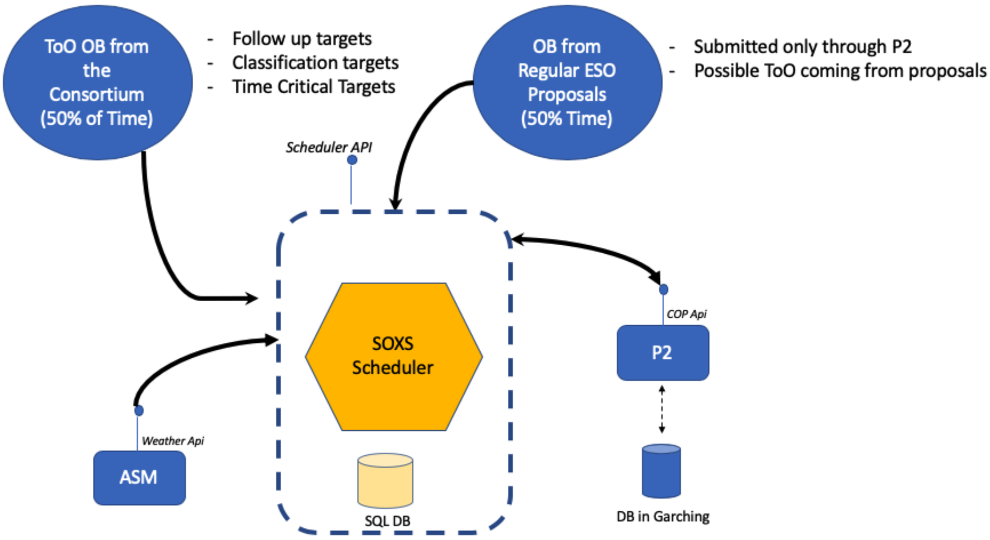


Figure 1 - SOXS Scheduler concept.

3.2 The Marshall Application

The Marshall Web Application (see Figure 2) will act as one of the two main sources of targets for the scheduler. It collects data from various surveys, aggregates them and produces a unique numeric ID, namely the ”ticket”, that identifies the transient. The Marshall also guarantees to provide the scheduler with the necessary updates on targets and we are adapting the already available and reliable Marshall for the ESO ePESSTO survey. The Marshall is an internal tool of the Consortium and cannot be accessed nor used by regular ESO users.

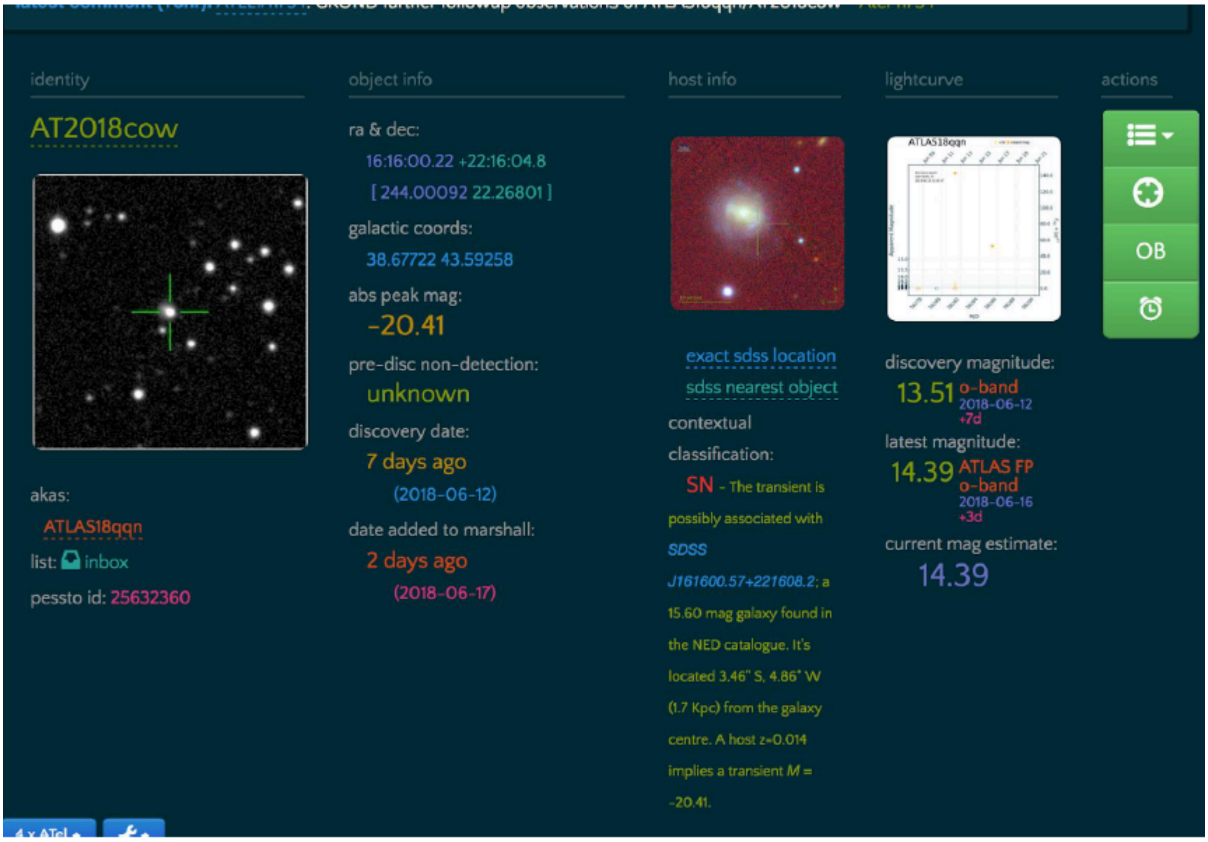


Figure 2 - An example of a Ticket in the ePESSTO Marshall (Young et al 2014).

3.2.1 Marshall features

Figure 3 reports the main logic of the source workflow in the SOXS Marshall. As explained in Section 3.2, the Marshall is intended to be used by the Consortium to aid in monitoring new and old transients. Its main features for the user will be to:

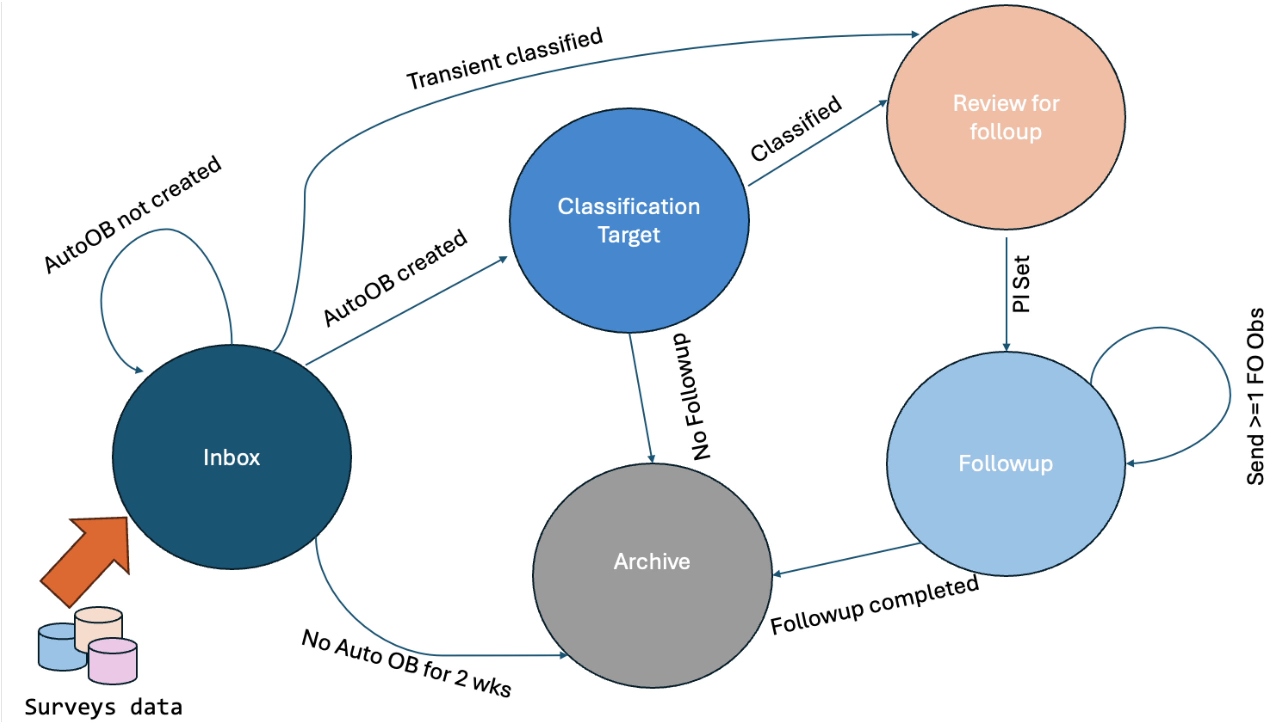


Figure 3 - Marshall source workflow.

- Keep track and continuously aggregate data from transient surveys (e.g. Rubin/LSST, ATLAS, etc.) of newly discovered objects (or, in general, transient events).
- Define a good set of targets, for which the classification is still unknown, to be followed up spectroscopically with SOXS (referred as Classification Targets).
- Mark some important targets for a dedicated Follow Up Campaign (in terms of spectroscopy) with SOXS (Follow Up Targets), with the ability to fully customize a related OB.
- Manually insert very urgent/particular objects (e.g. Gamma-ray bursts, or gravitational waves event) to trigger and follow observations with SOXS (UrgentTargets). Given the potential importance of these objects, the possibility of inserting them directly from the scheduler interface is also provided.

3.3 ESO P2 and Weather API

As discussed in the previous Sections, the main goal of the SOXS scheduler is to provide a plan for the night (validated by the Science Team before the night) and a back-up plan (according to the current weather conditions) by ensuring the right:

- Prioritization of the targets in terms of scientific returns and optimization of the use of the time at NTT.
- Managing of targets (mainly ToO) coming from both the Consortium SOXS Marshall and regular ESO-approved proposals.
- Simple interface for the operator on the mountain to recover a new OB to observe, in case of a sudden weather change, pointing restriction, or urgent OB.

The SOXS scheduler interfaces with the ESO P2 via the API tools provided, synchronises the approved schedule (initially only in the local database) with the database in Garching, checks and verifies OBs, and sends them one by one to the VES. This is a sorted list of OBs that can be fetched directly with the Broker of Observations (BOB) by the Telescope Instrument Operator (TIO) on the mountain, and it is fully integrated in the Control Software Environment provided by ESO. The Scheduler will only populate the VES of the Operator user. To achieve this, both the SOXS users and the regular ESO PIs must allow the Operator User to access their own OBs. This requires the use of the Phase 2 Delegation feature, which allows a user to gain access to the OBs from another PI in the P2 system. This is a key step for the Scheduler to populate the VES night by night and update it according to the current weather conditions.

4 Features and software architecture

The SOXS Scheduler will expose its services through RESTful API, and it will offer three main *macro-features:*

- A Web Application

  The Scheduler will implement a Web Application accessible through a normal browser that exposes two different levels of access.

- A Scheduler API

  A RESTFul API that exposes methods to create, update and manage the scheduler.

- A database

  A MariaDB database, used to store the data. A replica is foreseen also in La Silla with a dedicated machine to cope with sporadic network outages at the observatory. At the moment of writing, details are still to be discussed.

5 Scheduler functionalities

The SOXS Scheduler will expose its services through RESTful API that will be easily accessible by users through the use of a Web interface. The features developed for the Scheduler will be available through this Web application, and all the internal information will be stored in a dedicated MySQL database. The source code can be found at the github repository: https://github.com/marcolandoni/SOXSScheduler.
It is organized as follows:
The root directory contains the **docker-compose** files for deployment, the database initialization scripts under the **db**/ folder (including SQL migration files), and a **docs**/ folder with additional documentation.

The core of the application resides in the **app**/ directory, which contains the main script **(app.py**), auxiliary files for automated operations (**SOXSService.py** and its companion scripts for daemon operations, OB processing, and OB updates), and several standalone scripts for specific tasks such as finding chart generation (**fc.py**), observation updates (**update_obsob.py**), and six-month simulations (**six_month_simu.py**).

The **app**/ directory is further organized into the following submodules:

- **Filtering**/: implements the OB filtering logic and ESO grading system;
- **Instrument_utils**/: contains the SOXS instrument configuration;
- **P2_utils**/: provides utilities for interfacing with the ESO P2 system;
- **Scheduling**/: implements the core scheduling logic, including the night manager, the scheduling algorithm, and recovery loop utilities;
- **SQL_utils/**: handles all interactions with the MySQL database;
- **Telescope_utils**/: contains the telescope configuration parameters;
- **Utils**/: collects general-purpose utilities, including query functions, data storage, input validation, magnitude calculations, finding chart creation, and schedule visualization;
- **weather_utils**/: interfaces with the La Silla weather API;
- **config**/: stores the application configuration files (in YAML format);
- **simulations**/: contains utilities for running scheduling simulations;
- **logs**/: stores the application log files.

5.1 General overview

We first provide a general overview of the scheduler API, with a description of the main functionalities offered to create observing blocks, generate and modify a scheduler, and manage the observing nights.

OB creation:

- /**createAutoOB**: it creates an OB with information provided by the Marshall for a classification target. The only needed information are the coordinates in the sky and the latest known magnitude. Then, a default OB with spectroscopic Stare mode observations is created, with the exposure time depending on the magnitude and based on a lookup table;
- /**createFollowUpOb**: it creates a completely customizable OB, with information on the type of observations, observing times, number of exposures, weather constraints, time constraints, and others. All these information will be retrieved through the Marshall, where the Consortium Members can provide all these information;
- /**sendUrgentOb**: it creates an Urgent OB, that is completely customizable (as for the Follow-up OBs), and all the information are retrieved from the Marshall;
- /**create_ESO_OB**: it scans the P2 systems of delegated users with approved proposals to look for new requests for observations. Then, it create the corresponding OB with all the information provided by the users, leaving untouched the OB and only storing the corresponding information in the local database;

OB management:

- /**deleteOB**: deletes a specific OB when its is no longer needed;
- /**search_ob**: returns all information related to an OB, identified by its progressive number stored in the local database;
- /**modifyOB**: modifies a previously created OB. Any entry can be updated, including source information, the specifics of the observations, weather constraints and more;
- /**is_observable**: returns the observability status of an OB, identified by its progressive number stored in the local database;
- /**observable_obs**: returns all the observable OBs in the upcoming night;
- /**generateFindingChart**: generates a finding chart associated with a given OB, using a custom code and the standard ESO tools. The finding chart is temporarily uploaded to an online repository called Datalake;
- /**getTransientBucketIDOBs**: retrieves all the OBs related to a specific transient, all sharing the same TransientBucketID code.

Schedule creation and management:

- /**schedule**: creates the tentative schedule for the next observing night;
- /**suggestOtherOB**: suggests an alternative OB to potentially substitute another already scheduled, ensuring the same time slot is filled. If no OB with an equal

duration in found in the database, a shorter one may be suggested;
- /**substituteOB**: replaces a scheduled OB with another one. A substitution with a longer OB is not allowed;
- /**add_to_schedule**: add an OB to a schedule with available gaps. It is intended for manual scheduling only, this approach is not recommended (see below Section 7.4.10);

Night management:

- /**openNight**: open a new night session to start operations;
- /**closeCurrentNight**: close a previously completed night. This must be called in order to start a new night;
- /**gtoprogress**: displays the time used during the night, divided into observations for SOXS Consortium and ESO users;
- /**next_ob**: manages the night operations by selecting the next OB to observe, either when the current one is about to be completed or aborted, when or no OB is currently in observation (see Section 5.6).

5.2 Database

The API is currently capable of ingesting and locally storing data in a MySQL database from the Marshall (see Sect. 3.2), with a distinction between different types of targets, i.e. 'Classification' (/createAutoOB), 'Follow-Up' (/createFollowUpOB) and 'Urgent' (/sendUrgentOB) OBs. It also ingests OBs ('ESO' targets, /create_ESO_ob) provided by the community with approved regular ESO proposals. Classification targets are automatically provided from the SOXS Marshall, which in turn relies on information coming from several surveys, while Follow-Up and Urgent OBs are sent from members of the Consortium to observe already classified sources or newly-discovered transients, respectively.

Using the Python library *Astroplan* and two dedicated scripts, the scheduler filters the observable objects according to their visibility constraints and updates this data daily. Updates in the magnitude of classification targets from the Marshall and corresponding checks on the exposure time are taken care of as well (/UpdateAutoOBMagnitude). Follow-up targets can be completely customized by the user with constraint determination, time constraints, acquisition/science template choice, and specific parameters through the /createFollowUpOB endpoint. The existing Follow-Up and Urgent OBs can be viewed using the /getFUandUob endpoint. The /OB endpoint is used to view all the OBs, while /oneOB shows only the requested one. Any OB can be modified in every parameter with the /modifyOB endpoint. The deletion of an OB is executed by the /deleteOB endpoint. Upon using

/SaveToP2, the requested OB is synchronized with the P2 database.

Currently, the MySQL database is composed of tables. These tables are the following:
- *ob_list*, which contains all the necessary information regarding the target to be observed, e.g. coordinates, templates, weather/time constraints, etc.;
- *obs_bucket*, which contains all the information relative to the observability of each target in the *ob_list*;
- *schedule*, which contains the information regarding the schedule for the night;
- *nights*, that contain specific information regarding the observing night;
- *history_schedule*, which contains the information regarding the updates of the schedule during the night, if any;
- *failed_obs*, which contains the information of the OBs that have been marked as failed or aborted during the night;
- *average_conditions*, which contains the average weather conditions of the night at the beginning of each OB;
- *GTO*, which contains the information regarding the GTO time used by every user;
- *report*, which contains a list of necessary information for the report of the night;
- *soxs_USERS*, which contains a list of the SOXS users and their contact information;
- *daemon*, that simply stores the information linked to the running status of the next_ob_daemon (a script responsible for continuously inspecting the status of observation and providing the next OB).

There might be a few minor changes to the current table list based on some fine-tuning. We place a more detailed description of the contents of the database in the tables in the Appendix.

5.3 Filtering routines

The idea behind the SOXS scheduler is to optimize the time available during the night, given the observability of the targets, their priority and contingent variables such as those tied to weather conditions. This will be achieved by selecting objects that have a good chance of being observed during the chosen night, i.e. depending on their observability curve ("preliminary filtering", see Figure 4). We made use of the already existing, tested and trusted algorithms in the *Astroplan* library for Python, performing occasional tweaks and workarounds to suit them to our needs. This library allows us to compute the observability of a target with the chosen time grid, based on its coordinates on the sky, the selected time, date and year and the geolocation of the observatory on Earth. These allow us to work with static conditions as if the target were not a transient. In our filtering procedure, we implemented the *Airmass, Moon Separation, AtNight* and *Time* constraints. A check for the required fractional lunar illumination is also performed. In particular, the observations are scheduled within the astronomical night.

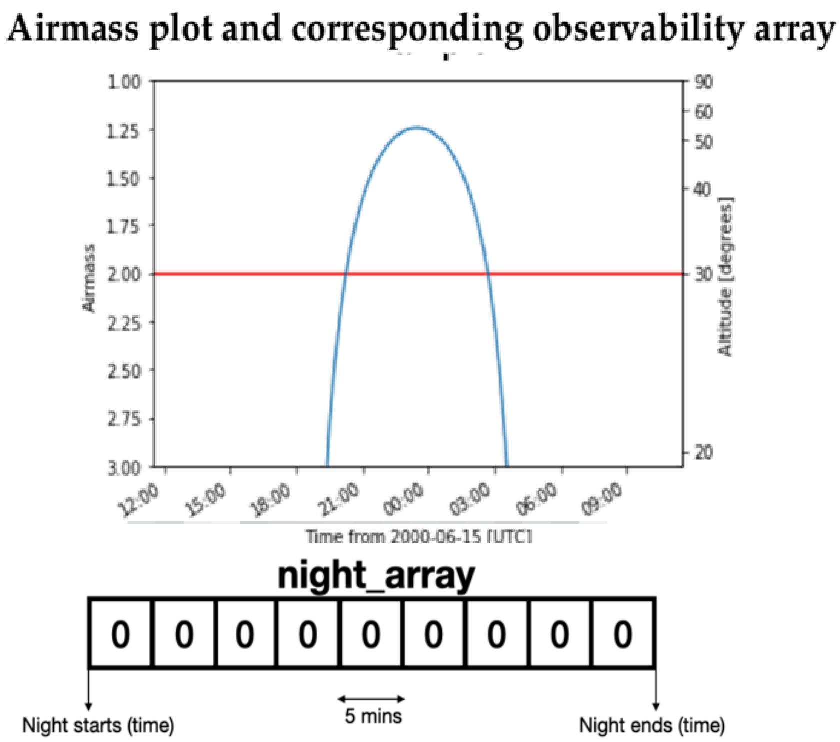


Figure 4 - Astroplan grid example.

Similarly to the *Astroplan* algorithms, we parametrize the night as a Boolean array with a number of slots ***n***, which is given by the length of the night, divided by the time grid we choose (5 minutes in our case, to account for overheads, Figure 4). The constraint evaluation made with *Astroplan* is then translated into a boolean array with n slots, the *observability_array,* which is the convolution of all of the chosen

constraints. The array presents a 0 if the object does not respect all of the specified constraints, and a 1 otherwise. This array is the basis for the scheduling algorithm we developed. The array is then attached to the corresponding OB for a given night in the *obs_bucket* table. Within the chosen grid, each target is also assigned a grade resulting from an extension we performed on the formulas currently used for target ranking in regular ESO operations. We started by computing the cumulative probability functions related to their observability, as described by Bierwirth T. (2010). Our grading system currently considers the probabilities related to airmass, sky transparency, seeing, fractional lunar illumination, and target setting. Each of these probabilities, which are evaluated based on the coordinates of the target and on the geographical specifics of the observatory, is normalized to one and then multiplied, giving the final grade to the target. The final value is expressed as a 3-digit string, allowing us to rank the targets. This result is evaluated in conjunction with the user priority assigned to each OB, i.e. if two targets have the same observability grade, they will be sorted according to the user priority. The user rank is assigned based on the importance and urgency of the OB through the Marshall or the ESO P2 system. User grades range from 1 (highest priority) to 10 (lowest priority).
To speed up the process and ensure that the scheduler always has the broadest possible view of future nights, we developed two algorithms that automatically process new OBs and update old ones. The first processing evaluates constraints for the current night and the following four nights. The updating script searches for OBs that have already been processed, slides the four nights back, and computes the new fifth night. These two scripts (*processing_obs.py* and *update_obsob.py*) run automatically and separately every 300 seconds, to ensure that every new OB ingested into the database is processed and existing ones are always kept up-to-date.

5.4 Scheduling algorithms

Before scheduling can begin, the science team has to open a new observing night (via the Web interface) after closing the previous one. Once a night is closed, it cannot be modified in any way, and no new OB can be sent to the VES. Besides, there can only be one night open at a time. After a new night is opened, the science team will ask the scheduler to propose a schedule. The parametrization of the night array described in Sect. 5.2 allows us to quickly run through the available slots and fill them within a few (~10) seconds.
To guarantee the maximum scientific output with the scheduler we developed, we split the algorithm in two, based on the priority of the targets.

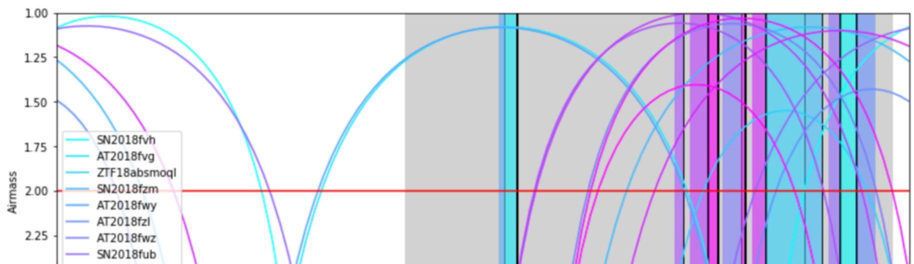


Figure 5 - Priority Scheduler example.

5.4.1 Priority Scheduler

To handle scientific prioritization of targets, particularly for Follow-Up Targets and ESO proposals, we take advantage of *Astroplan's* **priority scheduler**. It is designed to optimise observing time at one-minute resolution (see Figure 5), effectively maximizing the observing conditions. However, it has two notable limitations: computational intensity due to fine time resolution, and failure to utilise all available telescope time, often leaving significant gaps even when thousands of observable objects are available. Despite these drawbacks, tests confirm that the scheduler performs adequately for our expected workload, generating schedules within seconds and inserting targets with the best observability conditions into the proposed schedule. The output is formatted consistently with other database entries while retaining traceability to the scheduling algorithm.

5.4.2 Graded Gap Filler

After the priority scheduler has returned the list of prioritized objects, the night array is left with gaps of unused time that can be used by additional targets.
The scheduling process we developed (not only for Classification targets but also for emergency rebuilding of a schedule in case of disruption – whatever may be the cause) involves a further evaluation of the constraints for each target, and more importantly, it implements a competitive selection of the targets for each free slot of the schedule. During filtering, the object was associated with a grade, that parametrizes the probability of a good satisfaction of constraints.
For example, if the airmass limit of an object is 3, but the object does not rise above that airmass for long, and it always stays close to airmass 3, the object will have a low grade. On the opposite side, an object with the same constraint that rises up to airmass 1.5 and stays in the sky for hours will have a higher grade. This system is also useful to incorporate the scientific priority, and since it is sorted by grade, with

high-grade targets at the beginning, all that is required is to add a proper cypher in the slot of the tens. The algorithm (the **Graded Gap Filler**, or **GGF**, see Figure 6) looks for the free slots in the night array. The database presents the possible candidates from the database for filling that spot, sorting them by grade. The grade is evaluated taking into account some constraints, with the airmass as the most important parameter. A trace of which obs are already scheduled is being kept to avoid duplication problems, and the winner of the competition for each groupd of free slots in the night array is appended to the schedule and the night array is changed accordingly. A visualization of the Graded Gap Filler is given in Figure 6. This process is repeated until the night is full, if possible. Finally, the objects are sorted by time of observation, and an ordered list of targets is saved in the database, under the *schedule* table. Then, the Science Team can inspect it for approval or modification before the night starts.

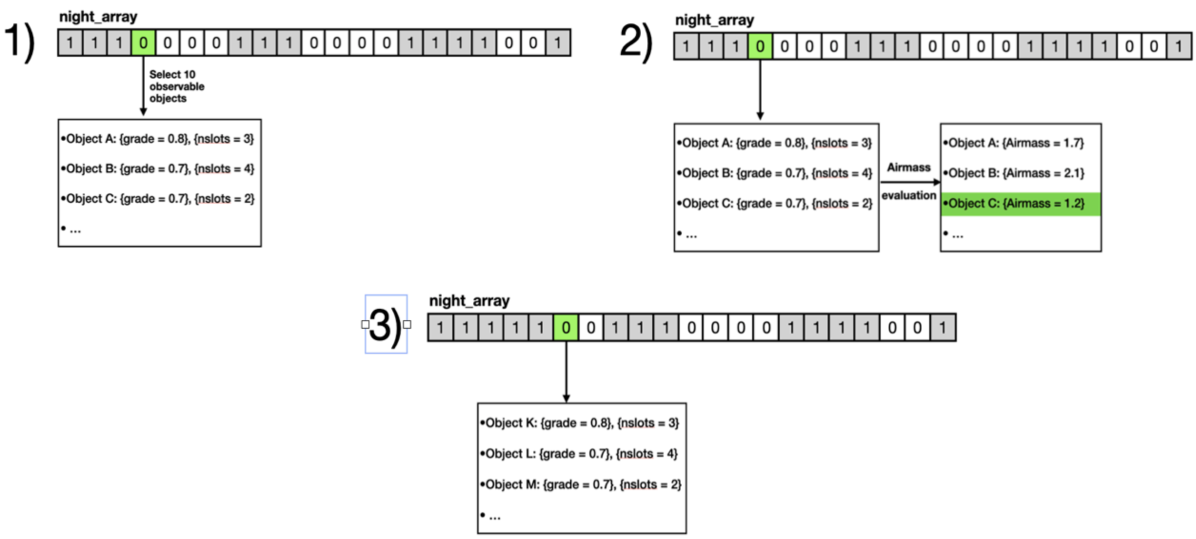


Figure 6 - Graded Gap Filler (GGF) algorithm schematic representation..

5.4.3 Priority and GGF in comparison

To test the effectiveness of the GGF for producing an optimized schedule that would indeed guarantee the good quality of observations, we used Monte Carlo simulations on the 2018 ePESSTO data, with these specifics:

- 40 different nights during the year
- 100 different schedules per night

With the following observability constraints:

- Airmass
- Fractional lunar illumination
- Moon separation

We computed the results as the difference between the median airmass during observation and the best airmass reached by the target during the night. For each schedule, we plotted both the results for the priority scheduler from *Astroplan* and those of the GGF. We found out that 90% of the events scheduled by the GGF were at less than 0.3 than their best airmass, which translates to about 2 degrees. Our results are displayed in Figure 7.

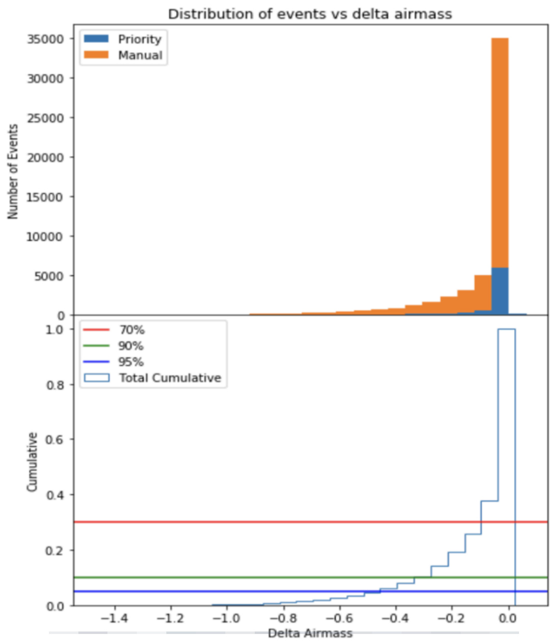


Figure 7 - Cumulative distribution of the events from their best observable airmass.

5.5 P2 synchronization and VES population

The SOXS scheduler interfaces with the ESO Phase 2 (P2) feature via the API tools, it synchronises the approved schedule (initially only in the local database) with the database in Garching (through the P2 system), checks and verifies OBs, and sends them one by one to the Visitor Execution Sequence (VES). This is a sorted list of OBs that can be fetched directly from the Telescope Instrument Operators (TIOs) at La

Silla Observatory. After the schedule has been approved by the science Team, the next action will be taken by the TIO during the night. In the dedicated interface, she/he will have to ask the scheduler for the OB to observe using the "next OB" button. This automatically prompts the checking process described in Section 5.6. Once the checks are successful, the OB is synchronized with P2 and pushed into the VES. The system is such that only one OB at a time can be in the VES. Then, the TIO only has to interact with the Broker Of Observations (BOB) to start new observations, with the BOB retrieving the OBs from the Visitor Operation Tool (VOT), which basically mirrors the VES in the P2. This procedure is fully integrated in the Control Software Environment provided by ESO. At the time of writing this document, synchronisation of the OB from the P2 to the Visitor Observing Tool (VOT) is not immediate, with the VOT retrieving information from the P2 every few (~5) minutes, slightly increasing downtime. This should not be a problem during normal operations, since the next OB is provided to the VES 10 minutes before the OB is completed. Only in the case of sudden changes (e.g. an OB failure, an Urgent OB), the VOT may not be updated immediately, requiring the TIO to click the refresh button to allow for synchronisation. After this, the VOT will be immediately updated, and the next OB can be fetched from the BOB.

5.6 Automated Night Management

The scheduler provides an algorithm that autonomously manages the observing night, both following the original, pre-approved schedule and reacting to any unforeseen circumstances.

It monitors and responds to weather changes and other real-time events occurring during observing nights. It updates the VES with a new OB whenever the previous one is about to be completed or has failed. The scheduler interacts with the weather API to retrieve the current visibility conditions and checks for time constraints and other requirements for the next OB candidate before approving and pushing it to the VES. This whole process is implemented through a 'daemon', i.e. a dedicated script that continuously runs (every minute, in our case), checking the current situation by tracking the current time, weather conditions, and OB status in the VES. The daemon runs continuously, but only acts during the astronomical night and if the triggering conditions are met (i.e. the previous OB is nearing completion, is aborted, or the VES is empty).

A dedicated endpoint (the "Next OB") takes care of the next OB selection, being designed to follow the original schedule or (if it is not possible) rearranging the schedule when necessary, preserving high-priority OBs and reoptimizing observations according to current conditions, both in terms of weather and time.

We identified 5 main categories of events that may require the triggering of a correction in the schedule:

- Delay in the execution of observations;
- Presence of previously failed observations;
- Advance in the execution of observations;
- Weather changes;
- Extra time left at the end of the night.

Each set of circumstances is handled by a dedicated "recovery loop", algorithms designed to address any issues that arise. As a guiding principle, we prioritise staying as close as possible to the original approved schedule by the Science Team, minimising changes. These loops together form the bulk of the “Next OB” endpoint, as shown in Figure 8.

Below we will give a schematic description of the workflow of this endpoint, while for the specifics on the Recovery loops we refer to the next subsections:

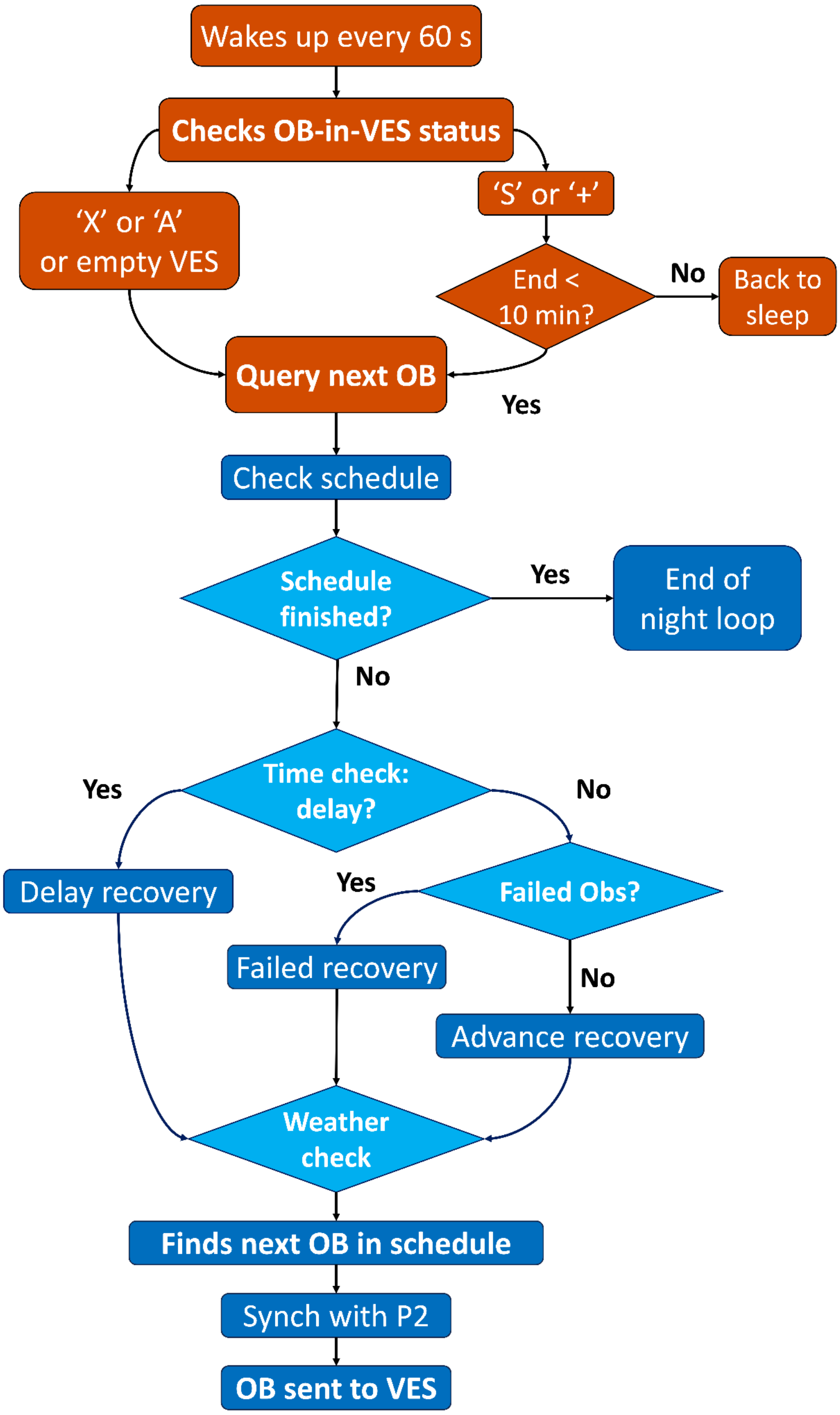


Figure 8 - Workflow of the next OB endpoint and the associated daemon.

1. Status Check: Verify the status of OBs in the present schedule, and select the first OB that is not marked as completed, in execution or aborted. It will be the next candidate to be added to the queue.
2. End-of-night Management: If we are close to the end of the night and the last OB in the schedule is close to being completed, check if any time will be left for observations. If it is the case, enter the End-of-night filling loop.
3. Time-constraint Check: Verify if the time constraints from the next candidate retrieved from the schedule (if any) are satisfied. If not, mark that candidate as failed and look for another candidate to fill the gap.
4. Delay Verification: Perform a time check to ensure no major delays have occurred. If detected, enter the Delay recovery loop.
5. Failed OB Check: If the schedule is on time, check the failed_obs table for any entries. If a failed OB for the current night is present, enter the Failed recovery Loop.
6. Advance Verification: Perform a time-check to ensure no major advances have occurred. If detected, enter the Advanced recovery loop.
7. Next OB pre-selection: Identify the next OB to be observed, either no modification is required or a new schedule is produced for the loops.
8. Weather Validation: Retrieve current weather conditions through the API and compare them against the next candidate OB's requirements. If requirements are not satisfied, enter the Weather Check loop.
9. Next OB confirmation: Identify the next OB to be observed, either no modification is required, or a new schedule is produced for the weather check.
10. P2 Synchronisation: Send the OB to the ESO P2 with all the information and details of the observation, synchronise it, and verify it.
11. OB Deployment: Send the OB to the VES for observation.

Each failed OB is either added to or removed from the table upon failing or observation. The table should be created upon opening of the night or upon schedule proposal.
Most importantly, before ANY rearrangement of the schedule, a function called *priority_check* makes sure that, upon rescheduling, all of the follow-up and high-priority OBs are maintained and rescheduled, if possible.

5.7 Recovery procedures

5.7.1 Delay Recovery loop

This loop (Figure 9) is used to check for the current time against the expected end of the observations. If the operations are late by more than a given time threshold (25 minutes in our case), the scheduler tries to re-adjust planned observations. The top priority targets are retrieved, run through the priority scheduler on the remaining time and rescheduled, with the new gaps created in the schedule that will be filled with the GGF. This process is nonetheless implemented to stick as much as possible to the pre-approved schedule. If any of the high-priority targets cannot be rescheduled, the schedule stays the same, and no modification occurs.

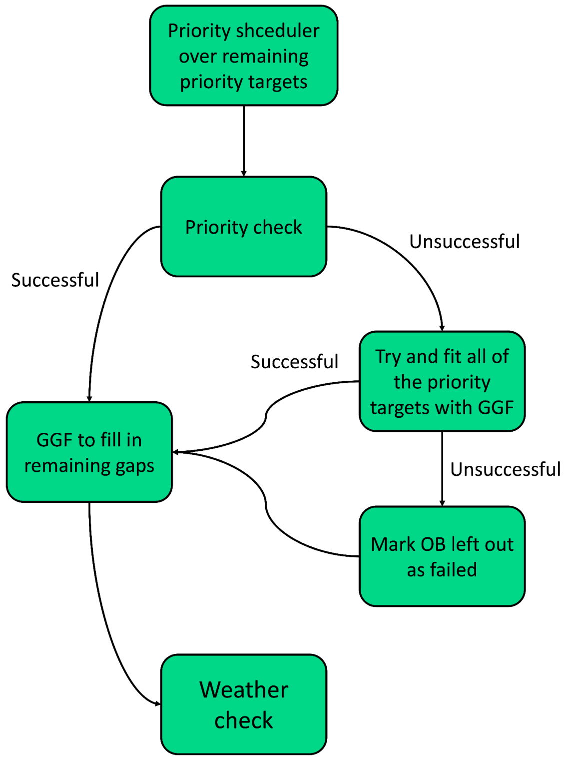


Figure 9 - Workflow of the delay recovery loop.

5.7.2 Failed OBs Recovery loop

It scans the table of failed OB on the night corresponding to today. For high-priority failed observations, an attempt to reschedule them in the remaining time is made, if the observability array allows it: the priority check is run over the remaining time to make sure that no other high-priority OB is discarded as a consequence of this addition. If the check is positive and the observability constraints are met in the available time slots, then the failed OB is recovered. Low-priority OBs that are left out as a consequence of this rescheduling are not marked as failed, as they are considered as "fillers", and the goal is to keep as many high-priority OBs as possible. Otherwise, the schedule stays the same. The workflow is shown in Figure 10.

5.7.3 Advance Recovery loop

Similarly to the Delay recovery loop, this checks for the current time against the expected end of the observations. In this case, if the operations (i.e. the expected end of observations) are early by more than a given time threshold (25 minutes in our case), the scheduler tries to fill this unexpected gap with the GGF. The rest of the schedule is left unchanged. If no suitable OB is found, we proceed with the previously approved schedule, and no addition is made.

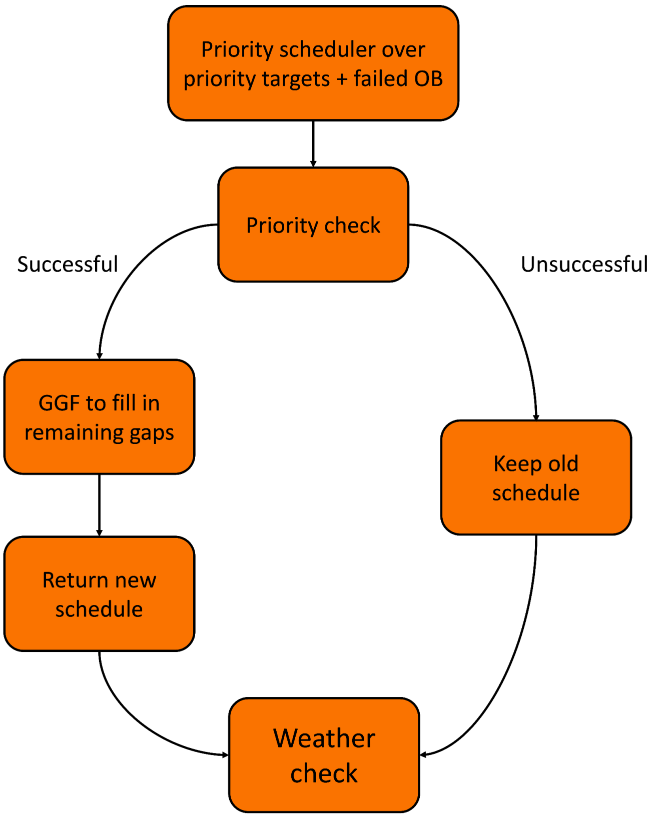


Figure 10 - Workflow of the failed OB recovery loop.

5.7.4 Weather check loop

This function compares the OB requirements with the conditions retrieved from the weather API. If the conditions are met, no modification occurs; otherwise, the OB is marked as failed, and we try to substitute it with the GGF. If successful, the new schedule is saved, and the OB is sent for observation. If no suitable OB exists, the weather conditions are incompatible with all remaining OBs for the night. Currently implemented weather conditions include: seeing, wind direction, wind speed, and airmass. The weather loop is run for every OB that is about to be observed, regardless of the other loops, and it is the last check before sending the OB to the VES for observations (see Figure 11).

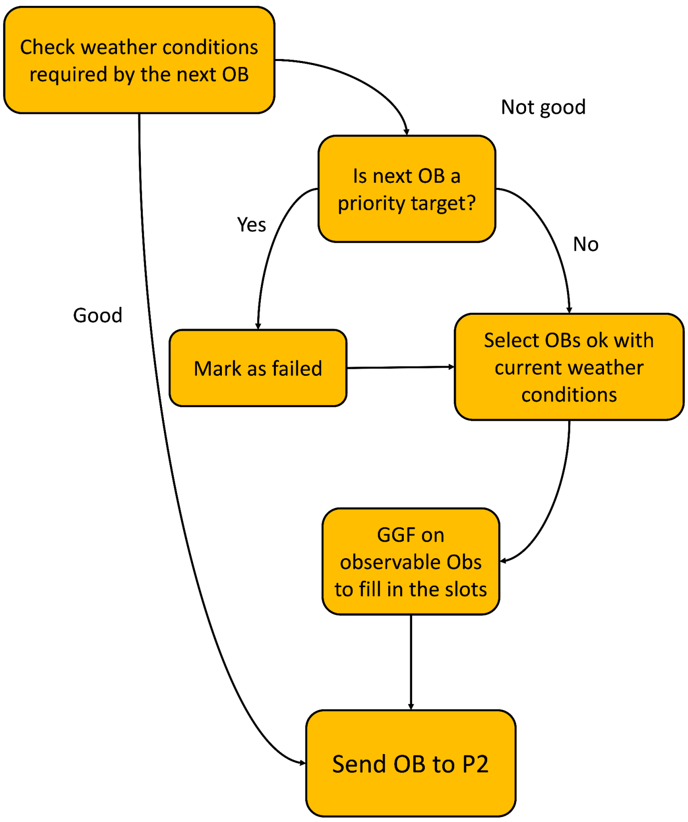


Figure 11 - Workflow of the weather check loop.

5.7.5 End of night loop

When the last scheduled OB is about to be completed (i.e., with less than 10 minutes of exposure time remaining), the scheduler evaluates whether sufficient time remains for additional observations before the end of the astronomical night.
The decision follows this logic: if fewer than 10 minutes remain until the end of the night, a standard star observation is executed. If 10 or more minutes remain AND no standard star has been observed it should be a standard practice in normal operations), the scheduler prioritizes observing a standard star to ensure photometric calibration for the night. If 10 or more minutes remain and a standard star has already been observed, the GGF attempts to fill the remaining time with an OB from our database. See Figure 12 for a schematic representation.

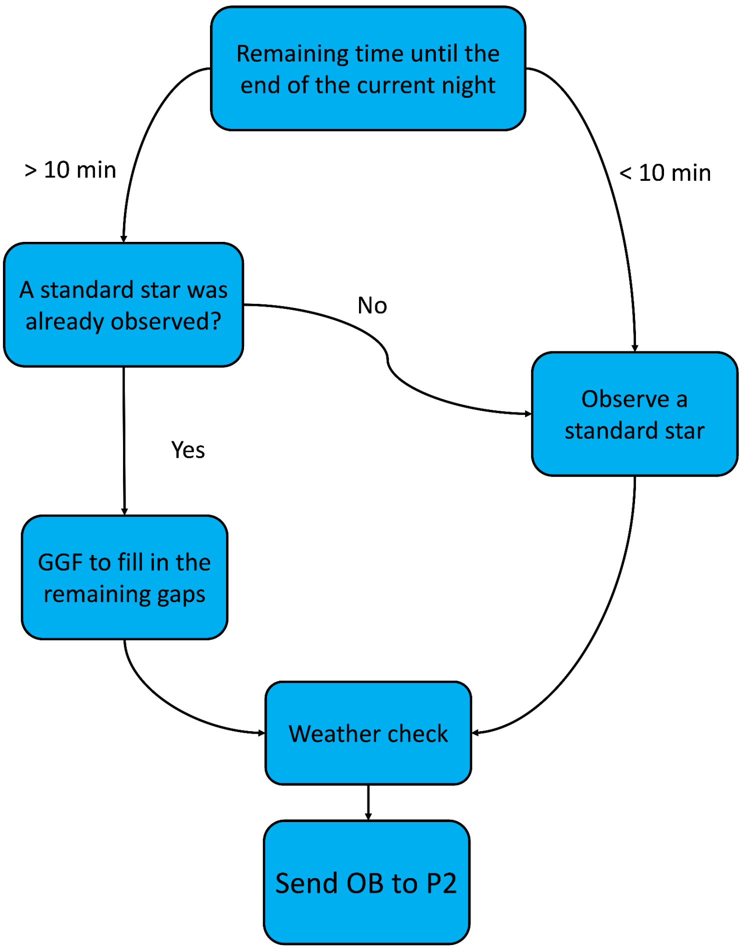


Figure 12 - Workflow of the end-of-nigth loop.

5.8 Other functionalities

In this section, we will discuss all of the functionalities currently implemented on the scheduler interface that we did not mention before.

5.8.1 ESO OB retrieval and P2 Delegation

The creation of OBs by ESO users with regularly approved proposals follows the standard ESO procedures. The user will use the Phase 2 system to create the OB (which can be fully customised), and will delegate either the TIO or someone from the SOXS science team. In this way, it will be possible for the scheduler to retrieve all information of every 'ESO OB' and ingest them into our local database. There is no agreement yet on the scientific priority to be assigned to every OB, therefore, they will be treated as high-priority targets, as for now. During the night operations, when an ESO OB is selected as the next OB, the synchronization/verification process in the P2 is skipped, and the OB is directly put into the VES.

5.8.2 Follow-up and urgent OB creation and visualization

While the creation of Follow Up OBs will be performed through the Marshall (for targets from the Consortium) or through P2 and subsequent delegation, the scheduler is prepared to receive and store these completely customizable targets. The scheduler keeps track of all OBs created for a specific transient coming from Marshall (transientBucketID). There is a way (\getTransientBucketIDOBs endpoint) which returns all the OBs associated with a particular transientBucketID (followup, urgent, classification, etc..).
Every follow-up OB, can be completely customized including information on weather constraints, if any:

- Acquisition template name
- Acquisition template details
- Science template name
- Science template details
- Observing constraints (such as airmass, sky transparency, seeing)
- Beginning of observation (optional)
- Ending of observation (optional)
- Fixed time observation (optional – default False)

In the night management window, a dedicated tab will show all of the follow-up and urgent OBs present in the database.

Urgent OBs are equally as customizable, but more information needs to be specified, such as the coordinates and all target info.

6 Performance

6.1 Simulations

To check the quality of the resulting schedules as a whole, we simulated three months of consecutive use of the scheduler, plugging in an artificial time passing, fake weather conditions and a flow of a thousand new targets per night. To keep the simulation as realistic as possible, we created the targets using the recorded transient alerts from the ePESSTO collaboration in 2018. To gain variety on the exposure times, we kept the coordinates and randomized the magnitudes, with 21 being the maximum allowed magnitude for SOXS. This way, the scheduler could work on 1000 new - potentially observable – targets each night. We then simulated the follow up requests the community will forward, both as SOXS Consortium and ESO users. These two groups have the same privileges, and they share 50% of the time at the telescope each. We created a number of these requests that was extracted by a Poissonian distribution with $\lambda = 5$, in order to have on average 10 requests each day.

Lastly, we extracted the randomized weather conditions following the published ESO data for Cerro Paranal (still in the Atacama desert). The “unforeseen events”, which can span from a telescope or instrument malfunction to an unregistered bad weather event, were all considered under the umbrella effect of a “delay”, that we set to range between half an hour to two hours and on average to occur every 2 days. The simulation then ran through most of the functionalities of the scheduler automatically. Each “day” started from the creation of the targets in the database (normal, Follow up and ESO), the database cleaned itself from the obs older than 30 days, and the filtering processes for the optimization of the target observation took place. Then a schedule was created, and a fake progression of the night cycled through the elements in it, considering the fake conditions and reacting to them. After one night, the whole thing repeats for the next “day”. During this, we saved the relevant parameters to evaluate the scientific throughput. The most telling parameter is once again the “delta airmass”. As Figure 16 shows, a satisfying 90-95% of the targets, which have all gone through the initial scheduling and the following night management process, are observed within a 0.2 airmass from their best, which for altitudes of about 45 deg is about a 3 deg vertical shift. The first schedule was observed with a mean delta airmass of 0.1 ± 0.2, that stayed substantially unchanged after the night management with a mean of 0.1 ± 0.1, even with the load being heavier on the GGF rather than the Priority scheduler. Real-time testing of the scheduler was performed during five nights in September 2024, using the EFOSC2 instrument. The results were overall satisfactory both in terms of algorithms (no faulty logic), and in terms of response time.

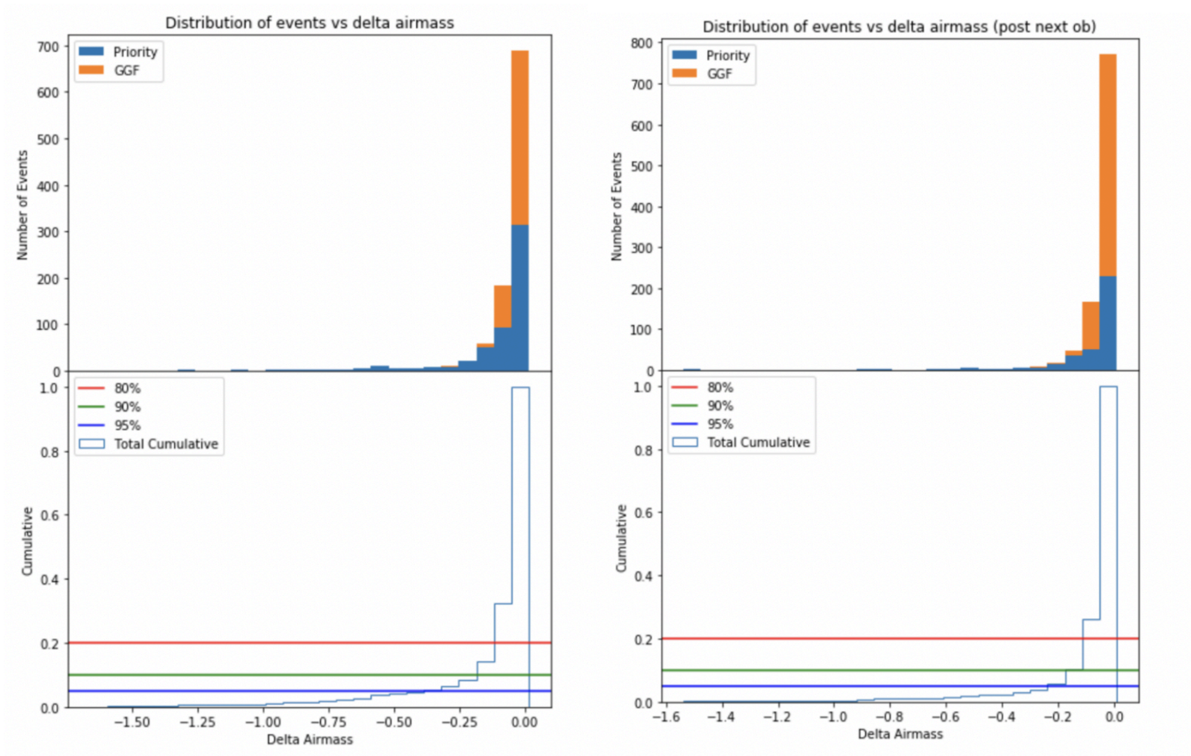


Figure 13 - Results of the simulations as described in Sect. 6.1.

6.2 On sky commissioning

During the last SOXS commissioning runs, extensive tests of the scheduler have been carried out since August 2025. The main goal was to ensure that the interaction between the scheduler and the ESO system is solid, and that the scheduler can successfully handle the sequence of observations during both a “regular” (no inconveniences) night and in case of unforeseen circumstances (e.g. delays in telescope opening, adverse weather conditions, etc.).

As anticipated in Section 1, the scheduler interacts with the ESO P2 system to provide the OBs to the VOT, then to the BOB, so that they are available to the TIO to handle observations.

6.2.1 Scheduler-P2 interaction

Starting from the OBs saved in our database, the scheduler can "translate" this information into a real Observation Block in the P2 system. In particular, we insert all the source info, template values, observing constraints and finding charts (those automatically created by ESO, those provided by the user - if any - and one created with a custom code). Once the OB is created in the P2 system with all this information, it is checked and verified. Then, it is sent to the visitor execution sequence (VES) in the P2, which corresponds to the VOT available to the TIO in the Control Room in La Silla. At this point, the TIO only needs to fetch the OB from the BOB and start the observation. This system was extensively tested and proved successful. We remind that, given the dynamic nature of the scheduler, an OB is sent to the VES (and the VOT) at most 10 minutes before the end of the previous OB, to ensure that the schedule can be dynamically updated, reacting to unforeseen events, if any. Therefore, without major problems encountered during the observing night, the next OB is provided 10 minutes before the end of the previous one, and it is available to the TIO to proceed with observations. The only operation requested from the TIO is to fetch the next OB from the BOB when the previous one has completed its execution. This procedure has been tested with success as well.

6.2.2 Recovery procedures

Should any inconveniences arise, the scheduler registers that an unexpected event happened and reacts to provide the next OB, adapting the schedule if needed. These inconveniences can come from a variety of different causes (e.g. operations are late, an OB is aborted during execution, the weather conditions prevent observation of a given target, etc), and a procedure to handle them is planned. The scheduler tries to keep all the 'priority OBs' present in the original schedule, keeping it as planned or generating a new one, if necessary. After that, the observations can continue with the newly provided schedule, as before. These procedures, which we described in Sect. 1.3, were tested during the commissioning runs.

In case an OB that is being observed is aborted more than 10 minutes before its end, the scheduler has not inserted the next OB into the VOT yet. Therefore, one minute, at most, is needed for the scheduler to register this unforeseen event and provide the next OB. In this case, the TIO needs to refresh the VOT page on its screen before fetching the next OB from the BOB. This happens because the VOT automatically refreshes every ~5 minutes to synchronise with the P2 system (see Sect. 1.5).

The end-of-night management has been implemented during the January 2026 run, being able to fill the end of the night, if any time is left before the end of the astronomical night. Both standard stars, if there was little time left or they had not already been observed during the night, and regular OBs, if a standard star had already been observed AND there was enough time left, were successfully scheduled and observed.

As a general conclusion, the scheduler interaction with the P2, VOT and BOB is stable, as confirmed by the test carried out during the commissioning runs. In particular, during the January 2026 run, some nights have been completely managed by the TIO present at La Silla Observatory (Cecilia Farias), under our supervision. This test aimed at simulating a real observing night during regular SOXS operations. The procedures ran smoothly, and the TIO was able to successfully fetch the OBs to be executed, recognise the field from the finding charts, and observe the sources correctly.

6.2.3 Standard stars observation

The management of observing nights also includes the observation of standard stars, in addition to scientific targets. This is required for calibration purposes. The plan is to observe a standard star at the beginning of each night. Therefore, the first ~10 minutes of each night will be dedicated to this kind of OBs. If this observation fails (e.g., the telescope is closed), a standard star observation is performed at the end of the night, or between two OBs if there is any gap in the new schedule. This may happen since the original schedule will likely be rebuilt in the case the operations start later than planned, and any 5-10 minute gap will be filled with this type of OB. Tests of standard star observations at the beginning, end and middle of the night have been carried out successfully during the January 2026 run.

6.2.4 Schedule creation and adaptation: OBs tested

Regarding the schedule creation, it is possible to request observations for both fixed and moving targets, with the latter requiring adequate finding charts from the users

to help the TIOs recognise the moving object, which is usually not as straightforward as a normal, fixed target. OBs for moving targets have been scheduled and observed with success three times during the January 2026 run.
In addition, the user can request a specific time range for their OB to be observed (e.g. for sources with peculiar emission during their orbital phases). This will be taken into account by the scheduler both when preparing the schedule and during the observing night. Therefore, if the requested time range is exceeded due to any delay, the observation will not be carried out. This procedure has been tested and proved successful. We also successfully tested the inclusion of "concatenated" OBs, i.e. one to be executed strictly after the other, to ensure longer exposure times for a single target (for example, for observations longer than 1 hr). This however only works for SOXS OBs, being this feature not enabled in the P2 in La Silla (so that an ESO user cannot build a concatenated OB).

In addition to that, should any new transient be discovered during the night (e.g. a gamma-ray burst, a gravitational wave counterpart, etc.), an "Urgent OB" requesting new observations can be sent. The on-call scientist for SOXS will send it directly to the VES/VOT. Therefore, the schedule will be disrupted to allow for this urgent observation and rebuilt at the following iteration. This procedure has been widely tested already during the September 2025 run.

We provide here a summary of the activities carried out during the January 2026 run. We first show a global summary of the statistics of the activities, comparing the number of OBs initially scheduled at the beginning of the night, those actually executed and the number of changes during the operations. We specify that in a few cases some tests of the instrument, not related to the scheduler, have been requested by ESO. Such activities will not happen during regular operations, but they have been a good way to simulate the procedure of schedule adaptation / disruption during an observing night because of general failures, bad weather, and any additional problem that may arise. This "stress test", after some corrections and improvements to our code during the first nights, was handled with a good performance by the scheduler overall. The number of changes were limited to a few per every nights, successfully keeping the OBs with the highest priorities untouched.

In the following table we report the general statistics of every night of operations during the January runs, then we show some specific examples of schedules, comparing the schedule originally proposed by the scheduler and the list of OBs actually executed during that night.

- n_OB_initial: Number of OBs present at the time of the first schedule (the one that should be approved by the Science Team during the European afternoon).

- n_OB_observed: Number of the OBs successfully observed that were present in the initial approved schedule.
- Removed: Number of OBs removed from the initial approved schedule
- Added: Number of OBs additionally observed during the night due to spare time gained by previously OBs not observed for weather conditions, instrument malfunctioning, telescope downtime, etc.
- Modified: Number of OBs successfully observed present at the time of the first schedule that have been observed with a delay higher than 2 minutes with respect to the estimated execution time in the first schedule.

night	n_OB_initial	n_OB_observed	removed	added	modified	notes
2026-01-09	8	3	5	5	3	First installation of the scheduler. Fixed problems with weather API.
2026-01-11	6	2	4	8	2	Aborted 1 Ob,Duncan training, template issues
2026-01-12	4	2	2	6	2	Template issues
2026-01-13	5	4	1	7	3	Unexpected error with the telescope
2026-01-14	9	8	1	1	4	Aborted 2 OBs,test ETC
2026-01-16	9	8	1	1	8	Aborted multiple acqs, rotator and VOT issues
2026-01-17	7	5	2	4	5	Aborted multiple acqs due to moving target
2026-01-19	7	5	3	2	4	Astigmatism analysis; Aborted

						2 OBs, multiple rotator failures
2026-01-20	8	5	3	2	5	Astigmatism tests, bogus and aborts due to rotator
2026-01-21	9	3	6	3	3	Opened late, problem with focus and bogus
2026-01-22	9	9	0	3	8	Aborted 2 obs in the beginning, moving target, error with template

In order to provide a better insight of the performance of the scheduler during the commissioning, we selected some representative nights that we describe below.

Night 14 to 15:

We report here a "normal" night of operations, beginning as expected and without major delays, i.e. not requiring a schedule adjustment. However, the second to last OB was changed as a part of additional tests for the "Urgent OB" procedure: we sent an OB (WDJ0648) simulating a request for an Urgent OB. It was successfully observed, the originally planned OB skipped, then the original schedule continued as expected since it did not undergo major delays.

INITIAL			FINAL		
Target	Start	End	Target	Start	End
EG21	01:18:38	01:28:37	EG21	01:45:00	01:50:35
AT2026ob	01:28:37	02:18:37	AT2026ob	01:53:22	02:29:06
4FGL_J0427.8	02:18:37	02:53:37	4FGL_J0427.8	02:31:52	02:41:18
SwiftJ0840.7	02:53:37	03:48:37	SwiftJ0840.7	02:53:33	03:48:27
SwiftJ0840.7	03:48:37	04:43:27	SwiftJ0840.7	03:48:47	04:43:22
SwiftJ0840.7	04:43:27	05:53:37	SwiftJ0840.7	04:43:23	05:53:38

AT2026ta	05:58:37	06:18:37	AT2026ta	05:58:42	06:18:37
SN2025ahkt	06:23:38	07:03:37	WDJ0648	06:23:38	07:03:37
GS1354	07:13:37	08:23:37	GS1354	07:27:46	08:30:32

--- STATISTICS ---

Planned OBs (initial schedule): 9

Initial OBs observed (in final log): 8

Initial OBs NOT observed: 1

Initial OBs with modified times (tolerance 5 min): 4

Planned night duration (initial schedule): 7.08 h

Total observed time (final schedule): 5.37 h

Efficiency vs planned night*: 75.8%

Effective open window (first to last OB in final schedule): 6.76 h

Efficiency vs effective open window*: 79.5%

**** We note that the overheads between the last detector readout and the next detector integration are not considered in the efficiency estimation. This includes the telescope preset, acquisition and target identification. For this reason, the actual efficiency is in principle higher.***

Night 15 to 16:

Operations started a bit later than expected because of a failure in the hardware (the mouse stopped working and we needed to reboot the whole system). Therefore, we skipped the standard star expected for the beginning of the night and continued with the schedule untouched. No major delays were encountered, then no modification where needed. Having a few minutes left at the end of the night, the standard star was added in that spot, to complete the needed observations and properly calibrate the spectroscopic observations taken. This also shows the procedure of end-of-night filling in the case a standard star was not observed at the beginning of the operations.

INITIAL			FINAL		
Target	Start	End	Target	Start	End
EG21	01:18:14	01:28:14	WDJ0222	01:45:22	02:28:05
WDJ0222	01:28:15	02:23:15	QSO03188	02:28:55	03:38:08
QSO03188	02:23:14	03:33:14	09040_NGC3201	03:39:23	03:41:43
09040_NGC3201	03:33:15	03:53:15	SN2025aggl	03:47:50	04:33:33
SN2025aggl	03:48:15	04:48:15	SN2025advo	04:36:23	05:14:18
SN2025advo	04:58:15	05:43:15	G586825573541329536	05:22:16	06:23:12
G586825573541329536	05:43:15	06:58:15	SN2026acd	06:25:18	07:07:33
SN2026acd	06:23:15	07:18:15	07952_NGC3201	07:09:46	07:54:03
V462Lup	08:03:15	08:23:15	V462Lup	07:59:52	08:17:40
			LTT3218	08:27:50	08:46:33

--- STATISTICS ---

Planned OBs (initial schedule): 9

Initial OBs observed (in final log): 8

Initial OBs NOT observed: 1

Initial OBs with modified times (tolerance 5 min): 5

Planned night duration (initial schedule): 7.08 h

Total observed time (final schedule): 6.36 h

Efficiency vs planned night*: 89.8%

Effective open window (first to last OB in final schedule): 7.02 h

Efficiency vs effective open window*: 90.7%

Night 18 to 19:

We show here another night with a good performance of the scheduler, and again the end-of-night procedure. In this case, a standard star was already taken at the beginning of the night but only 10-15 minutes were left at the end, therefore no scientific observation fit that gap, and an additional standard star with a different airmass condition was acquired.

INITIAL			FINAL		
Target	Start	End	Target	Start	End
GD71	01:16:54.	01:31:54	GD71	01:20:27	01:35:44
SN2026ob	1:26:54	02:16:54	SN2026ob	01:46:02	02:22:22
Campo 6	02:11:54	02:31:54	Campo 6	02:25:07	02:39:01
Campo 4	02:26:54	02:46:54	Campo 4	02:40:10	02:53:00
ATLAS26aeu	02:51:54	03:11:54	ATLAS26aeu	03:13:15	03:19:00
4FGL_J0427.8-6704	03:26:54	04:41:54	4FGL_J0427.8-6704	03:30:00	04:36:09
4FGL_J0427.8-6704	04:36:54	05:51:54	4FGL_J0427.8-6704	04:38:18	05:42:39
4FGL_J0427.8-6704	05:46:54	07:01:54	4FGL_J0427.8-6704	05:45:04	06:49:01
SN2025ahag	07:01:54	08:07:00	SN2025ahag	07:02:08	08:07:13
			LTT3218	08:19:30	08:33:49

--- STATISTICS ---

Planned OBs (initial schedule): 9

Initial OBs observed (in final log): 9

Initial OBs NOT observed: 0

Initial OBs with modified times (tolerance 5 min): 5

Planned duration (initial schedule): 6.84 h

Total observed time (final log): 5.97 h

Efficiency vs planned night*: 87.2%

Effective open window (first-last OB in log): 7.22 h

Efficiency vs open window*: 82.6%

Night 19 to 20 (Cecilia e Felipe):

We tested a whole night managed by the TIOs (Cecilia and Felipe), trying to simulate normal operations. We started more than an hour later than expected because of astigmatism analysis operated by ESO. Therefore, the first OBs were skipped and we started rebuilding the schedule starting from ~2.36 UT. The majority of OBs originally scheduled had high priority (and/or high grade for their observability), so we only shifted a little bit the observing times. There was a failure with the rotator and the TIOs had to manually recover the telescope. However, an OB was skipped because it would have ended too late in the night (GS1354-645), and the corresponding empty slot was filled with a different target. The night then ended with a standard star since at the beginning we needed to skip it, having started operations too late in the night.

INITIAL			FINAL		
Target	Start	End	Target	Start	End
EG21	01:16:24	01:26:24	4FGL_J0427.8-6704	02:36:30	02:39:12
Campo 4	01:26:24	01:46:24	4FGL_J0427.8-6704	03:40:15	04:45:22
EG21	01:46:24	01:55:24	4FGL_J0427.8-6704	05:28:01	06:29:41
4FGL_J0427.8-6704	02:11:24	03:26:24	SN2025aggl	06:30:16	07:17:20
4FGL_J0427.8-6704	03:26:24	04:36:24	GS1354-645	(aborted)	-

4FGL_J0427.8-6704	04:36:24	05:46:24	WD_1056-384	07:43:17	08:07:42
SN2025aggl	06:06:24	07:06:24	LTT3218	08:12:28	08:28:48
GS1354-645	07:11:24	08:31:24			

--- STATISTICS ---

Planned OBs (initial schedule): 8

Initial OBs observed (in final log): 5

Initial OBs NOT observed: 3

Initial OBs with modified times (tolerance 5 min): 4

Planned duration (initial schedule): 7.25 h

Total observed time (final log): 3.62 h

Efficiency vs planned night*: 49.9%

Effective open window (first-last OB in log): 5.87 h

Efficiency vs open window*: 61.7%

Night 20 to 21:

Operations started later than expected because of astigmatism analysis operated by ESO. As above, the initial targets (standard star included) were skipped, then a few adjustments were made, trying to keep the priority targets and filling the empty slots with additional sources, or slightly moving the original observing times to fill the whole schedule.

INITIAL			FINAL		
Target	Start	End	Target	Start	End
GD71	01:15:52	01:30:52	SN2025pht	03:07:56	03:48:43
Campo 4	01:30:52	01:40:52	SN2025aggl	03:57:55	04:43:43
SN2025pht	01:40:52	02:35:52	SN2025advo	04:44:31	05:22:27

AT2023aefi	02:35:52	03:25:52	AT2026auc	(aborted)	-
SN2025aggl	03:25:52	04:25:52	G3846850989 822335488	05:50:31	06:52:00
SN2025advo	04:25:52	05:35:52	GS1354-645	07:09:10	08:23:43
G3846850989 822335488	05:55:52	07:10:52	LTT3218	08:29:02	08:44:15
GS1354-645	07:15:52	08:30:52			

--- STATISTICS ---

Planned OBs (initial schedule): 8

Initial OBs observed (in final log): 5

Initial OBs NOT observed: 3

Initial OBs with modified times (tolerance 5 min): 4

Planned duration (initial schedule)*: 7.25 h

Total observed time (final log): 4.60 h

Efficiency vs planned night*: 63.4%

Effective open window (first-last OB in log): 5.61 h

Efficiency vs open window: 82.0%

Night 22 to 23:

During this night operations began on time, but some problems were encountered trying to acquire the first OB. In this case, we tested additional templates and found some errors (then fixed) both in the templates and in the info provided by the users. Therefore, after a few tries, we gave up and moved on to another target, with the schedule that was adjusted to fill the empty slot. Then, we continued as planned and we managed to observe a couple of moving targets, as part of the expansion of possible OBs to be managed by the scheduler. We also tested a procedure during the schedule creation that put the standard star at the end of the night only. It will be useful in the case a target is observable only at the very beginning of the night (as AT2021afpi was) and we cannot lose time for other kinds of observations.

Despite the problems encountered trying to observe the first OB, it was a useful debugging test, also for the information we need for the users who send their request for observations, especially in peculiar cases and many different specifics for the OBs to be executed.

Target	Start	End	Target	Start	End
AT2021afpi	01:14:42	03:04:42	AT2021afpi	(aborted)	-
SA98-978	02:54:42	03:09:42	AT2025almw	(aborted)	-
NEO500080	03:15:50	03:45:10	3FGLJ0627.9-1517	02:15:41	02:46:28
AT2026ayt	03:44:42	04:24:42	SA98-978	02:49:41	02:56:26
SN2026A	04:34:42	05:04:42	NEO500080	03:21:23	03:42:12
GS1354-645	05:24:42	06:49:42	AT2026ayt	03:42:28	04:18:22
NEO523604	06:49:50	07:14:10	SN2026A	04:20:18	04:49:57
SN2026kc	07:15:42	07:49:42	GS1354-645	04:52:05	06:01:27
SN2026acd	07:49:42	08:34:42	NEO523604	06:03:17	06:27:12
			SN2026kc	06:28:49	07:04:10
			SN2026acd	07:09:28	07:38:39
			LTT3218	07:40:12	08:31:07

--- STATISTICS ---

Planned OBs (initial schedule): 9

Initial OBs observed (in final log): 9

Initial OBs NOT observed: 0

Initial OBs with modified times (tolerance 5 min): 5

Planned duration (initial schedule): 7.33 h

Total observed time (final log): 5.46 h

Efficiency vs planned night*: 74.5%

Effective open window (first-last OB in log): 6.26 h

Efficiency vs open window*: 87.3%

6.2.5 Templates tested

Observations with different templates have been completed during the commissioning runs, including:

- Soxs_slt_acq
- Soxs_img_acq
- Soxs_img_obs
- Soxs_slt_obs_Stare
- Soxs_slt_obs_AutoNodOnSlit
- Soxs_slt_Obs_GenericOffset
- Soxs_slt_cal_SpecphotStdStare
- Soxs_slt_cal_SpecphotNod
- Soxs_slt_cal_TelluricStdStare
- Soxs_slt_cal_TelluricStdNod

Several types of transient sources have also been observed, e.g. supernovae (belonging to different classes), cataclysmic variables, X-ray binaries, AGNs, QSOs, Novae, RR-Lyrae, and asteroids.

6.3 Shortcomings and problems encountered during the commissioning runs

We encountered a problem when creating a new OB, both from the scheduler and from the P2, when keeping the default values. The number of exposures for the UV-VIS is automatically set to 2 instead of 1. This is a software issue that was already reported by Davide Ricci and has now been solved.

6.3.1 Internet connection down in La Silla

A possible issue may be related to the absence of an internet connection in La Silla. To ensure the possibility of continuing observations, we will provide the schedule, approved by the science team, to the operator in La Silla before the start of the night, with all the OBs already created in an offline folder. In this way, they can retrieve every OB and continue with operations. In this case, no dynamic adaptation of the schedule will be possible, and the operator has to follow the observation sequence provided at the beginning of the night. However, this internet connection failure is expected to happen very rarely in La Silla, and this backup plan would allow normal operations to be carried out.

6.3.2 Improvements

The core of the scheduling processes and interaction with the ESO system is now settled. We plan to further improve the scheduling procedure, e.g. to include a system of scientific priorities for the targets requested, or to refine the selection of candidates (the “classification” OBs) to fill the gaps of the schedule made up of the priority targets. This procedure is independent of the night operations and interaction with ESO or the telescope in La Silla. We plan to have these improvements ready in the forthcoming months and further refine them after a few weeks of regular activity.

7 Web Interface

This section described the structure of the web interface used to interact with the SOXS Scheduler API.
Major information can be found in the AD7 document.

7.1 Interface software Structure

The following sketch illustrates the interface structure.

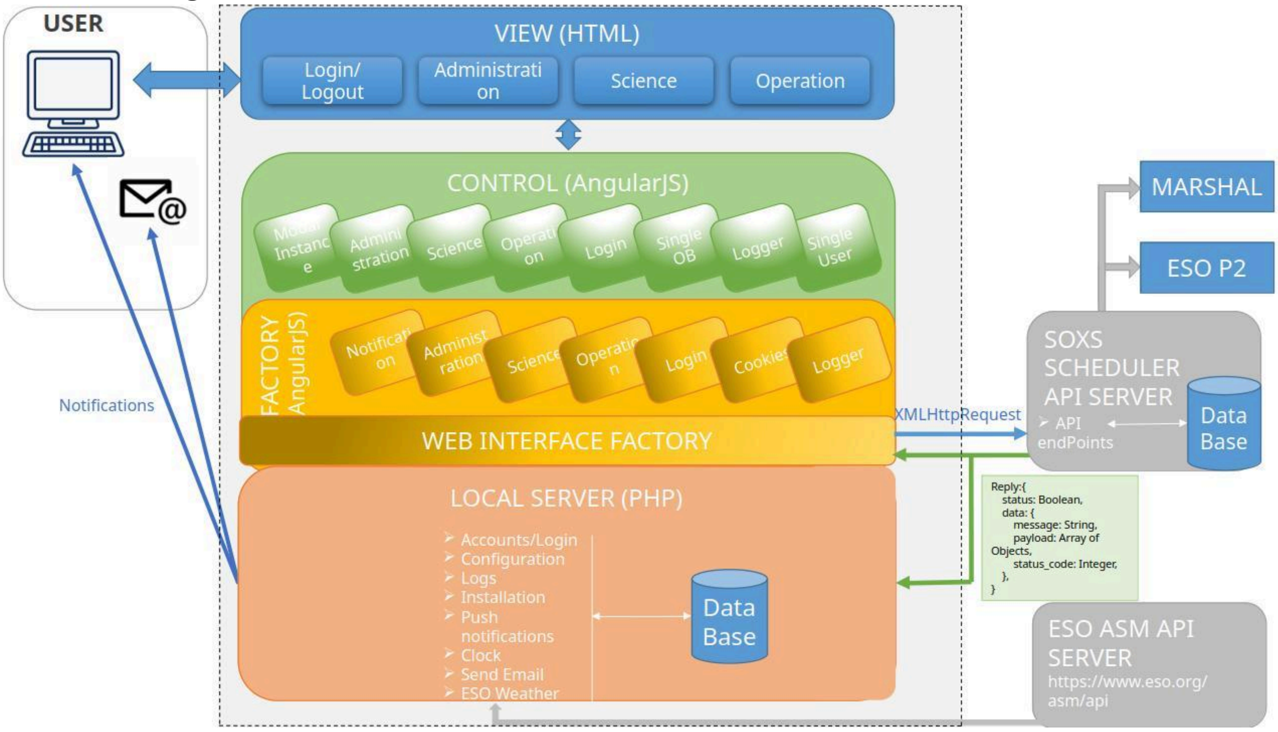


Figure 14 - Interface structures.

The Soxs Scheduler interface utilizes the MVC framework (Model-View-Controller) to decouple data management, user interface logic, and the visual presentation layer.

1. **The View (Presentation Layer)**

   This layer represents the front-end interface. It is responsible for rendering the visual elements that the user interacts with.
   The application opens on the login form, where different users from specific groups can access different sections of the interface. We defined 3 groups of users:
   - Administration: has the ability to configure applications, like modify the URL of the API server, and the user management (add/remove user). They also have access to the complete log of the application activity, that other groups have limited access to.
   - Science: as mentioned before, activities of this group of users is the organization of the night's schedule and of the visitor execution sequence.

This group has access to the log of the application activity generated by the Science team and by the operators.

- Operation: this group is active during the night and monitors the scheduled observations. It has access only to the operator's activity in the application logs.

2. **The Controller (Logic Layer)**
   Powered by **AngularJS**, serves as the intermediary between the View and the Model.
   It consists of specific logic modules that listens to events from the View, processes the input, and dictates how the Model should update.

3. **The Model (Data & Logic Layer)**
   The Model is represented by the AngularJS Factory and the Local Server (PHP/Database). It manages the data logic, performs calculations, and interacts with the Database and external API Servers (such as the SOXS Scheduler API and ESO ASM API). When data changes within the Model, it notifies the Controller so the View can be updated accordingly.

The model interfaces with three main online services:

- A Local Server, implemented in PHP and available on the same machine. It stores data required or produced by the interface. These are configuration files, user details and credentials and activity logs.
- The Scheduler API.
- The ESO ASM API Server that provides Weather data to the application

An HTTP interface factory (webIntefaceFactory.js) centralizes all functions that forward requests to the API server and to the Local Server, using AngularJS $http service (based on XMLHttpRequest).

7.2 Brief view description

The interface is divided into four sections:

- **Top Frame**: Displays current connection status with the scheduler (a green led that turns in blinking red if connection is lost), the current UTC clock (synchronized with the scheduler clock) and the logged-in user's menu.
- **Left Frame**: Contains the main menu specific for a group
- **Central Frame**: Displays the active page for user interaction.
- **Bottom Frame**: Contains copyright information and a link to the SOXS website.

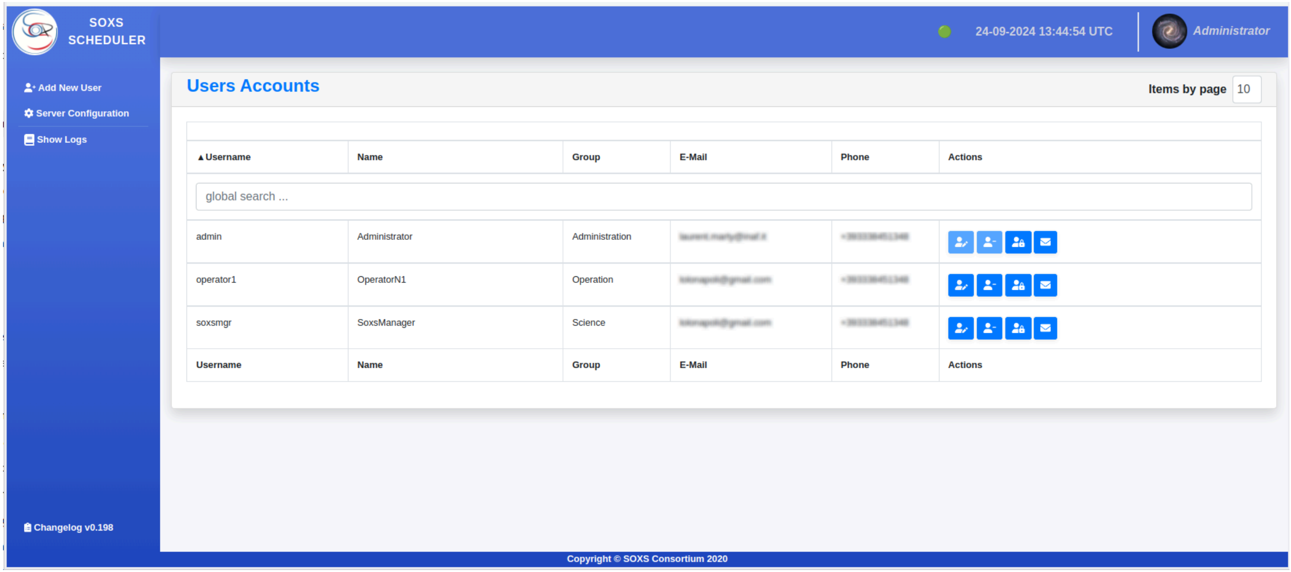


Figure 15 Soxs Scheduler Interface view

7.3 File and directories structures

All the stylesheets and libraries required by the application are included in the file index.html. It contains in its body a wrapper that is redirected using the angularJs routing service to the desiderated page.
A sketch of the structure of the application is shown in Figure 16.
The figure 17 shows the directory structure

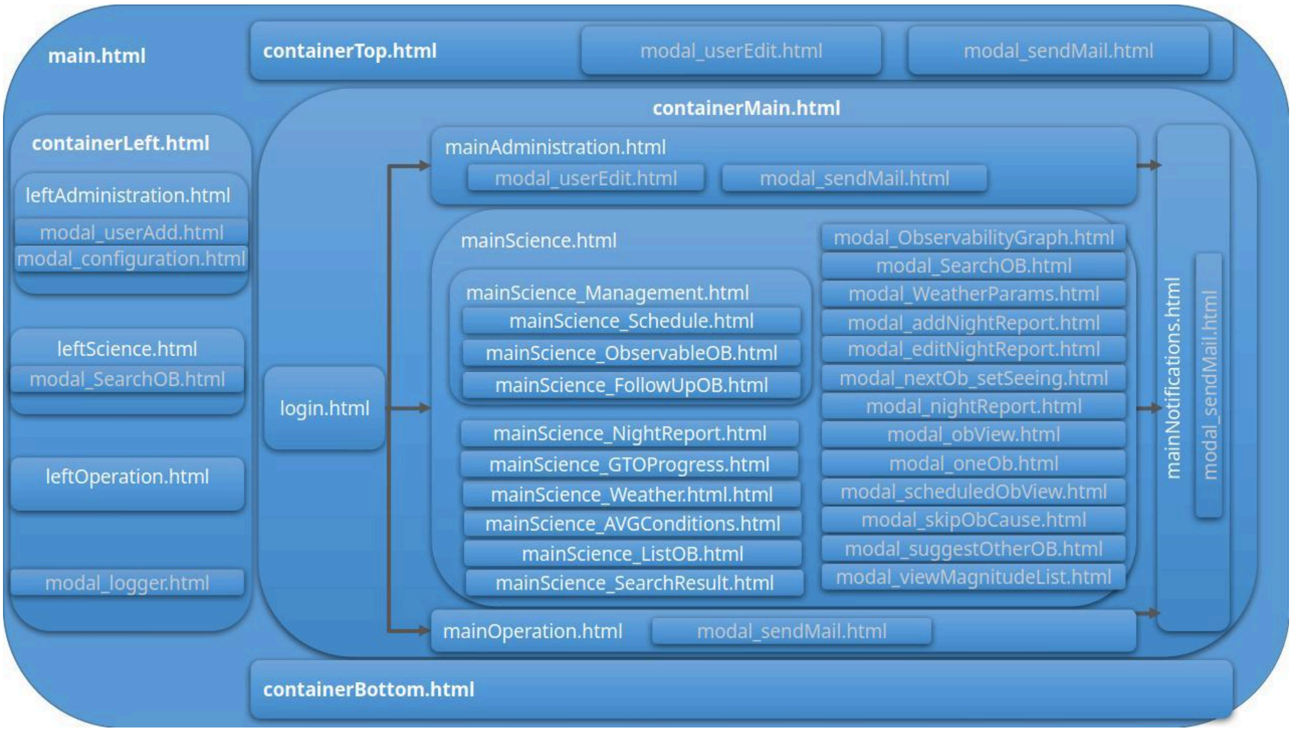


Figure 16 - Structure of the application.

```
.                        SOXS Scheduler Interface root directory
├── _localServer         ├── Local Server root directory (PHP)
│   ├── PHPMailer        │   ├── Library used to sent email
│   ├── class            │   ├── Class definitions
│   └── sql              │   └── Backup of database
├── css                  ├── Definition of the visual design of the site.
├── documentation        ├── Contains useful reading like this document
│   └── html             │   └── Auto-generated documentation
├── html                 ├── The View part (pages)
│   ├── administration   │   ├── Administration group pages
│   ├── modals           │   ├── Pop-ups
│   ├── operation        │   ├── Operation group pages
│   └── science          │   └── Science group pages
├── images               ├── Illustrations use in the site
├── installation         ├── Installation files (html & js)
├── js                   ├── JavaScript used
│   ├── controllers      │   ├── AngularJS controllers
│   └── services         │   └── AngularJS services
└── lib                  └── Third-party Libraries
    ├── css                  ├── css
    ├── fonts                ├── fonts
    ├── img                  ├── img
    └── js                   └── js
```

Figure 17 - Directories Structure.

7.4 Login and accounts

The user accounts are stored in a local MySql database. They are accessed through requests to the local server by the webInterfaceFactory. The Login form uses the login controller to check the userId/password inserted and reject or accept the login, rerouting the user to the right page of their group membership:

- Administration
- Science
- Operation

Members of the administration group are in charge of the creation of users, but each user has the possibility to modify this information.
Password hashing uses the bcrypt function.

7.4.1 Administrator interface

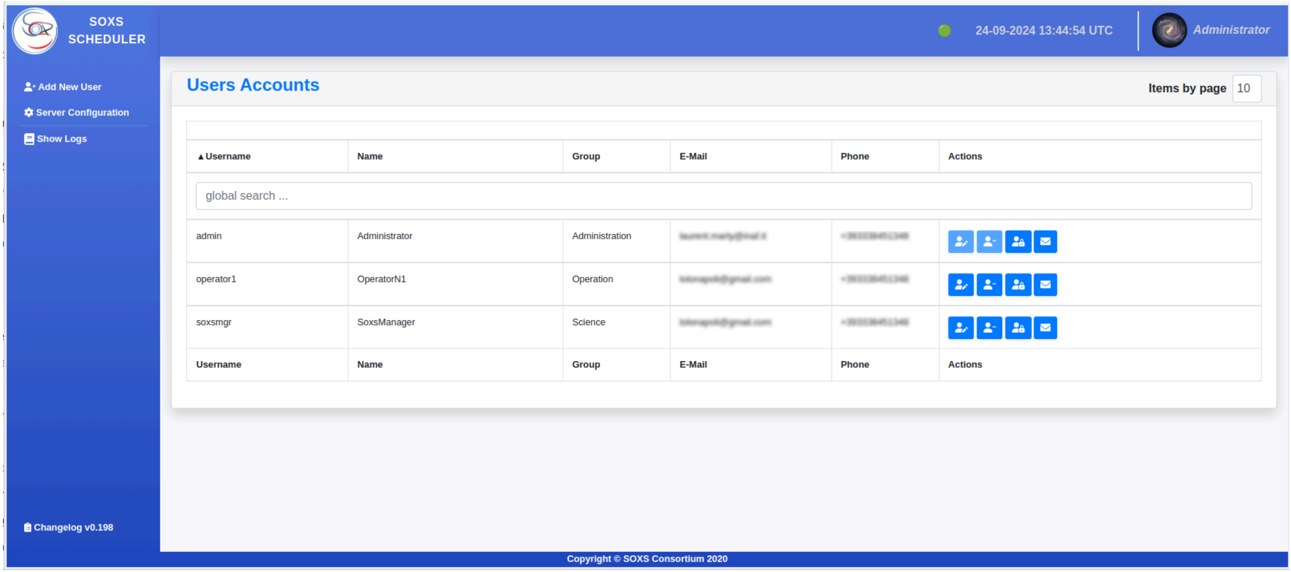


Figure 18 - Administrator Interface.

The administrator page shows a sortable, searchable table that lists all of the users accounts that are present in the system. From this page the administrator can manage these accounts:

- Add/edit/delete an account
- reset an account password
- send emails to a specific users

Further, from this page, an administrator can configure some setting of the application, such as:

- the Scheduler API server URL
- the duration of a session before an automatic logout
- the credentials used to access the SMTP service used by the scheduler to communicate to users

The administrator can also view all the activity logs from all groups and has the privilege to cancel all the log entries.

More information in document AD7

7.4.2 Science team interface

As mentioned before, the science team is in charge of asking the scheduler to propose a schedule, and thus to kickstart the operations during the night. The interface opens on the night management page, shown in Figure 19, with the previous night still open. The first action needed is to ensure that the current night is closed and open a new one. As for the other possibilities the science team has, they are discussed below.

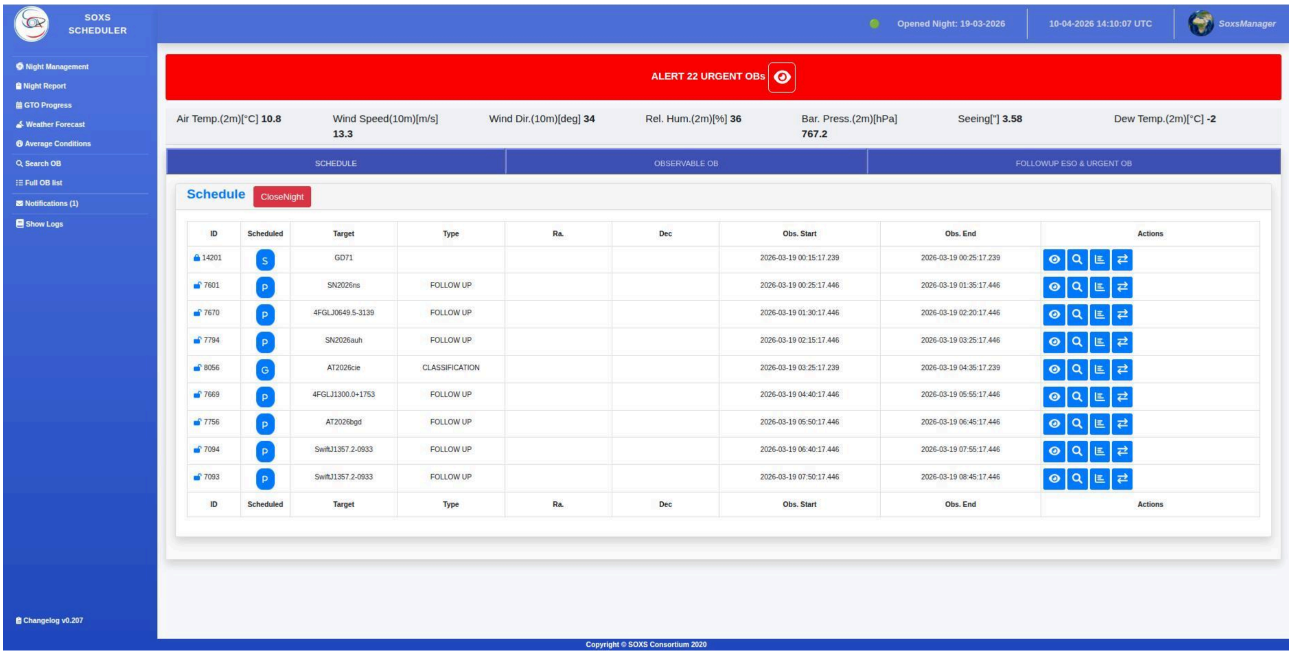


Figure 19 - The opening page, after login, for the science team group.

More information in document AD7

7.4.3 Operator interface

The operator interface will show the current VES, with notifications from the observers (in case a Urgent OB arrives), and with the ability to send reports during the night. Figure 20 shows a prototype of this page. The operator will also have the possibility of manually inserting weather parameters in case of failure of the ASM API, and it will show the current status of the daemon (running, not running) and of the internet connection (working, not working).

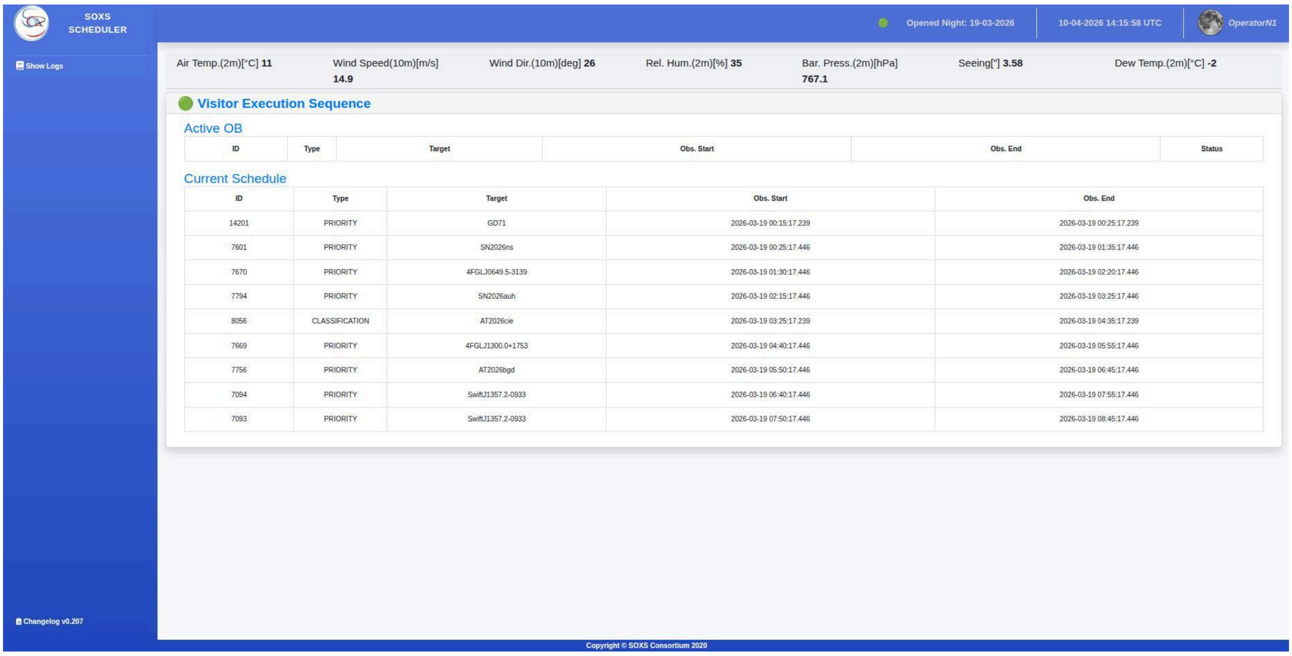


Figure 20 - Prototype of the operator interface.

More information in document AD7

7.5 Science team operations

7.5.1 Actions on OBs

The actions the user can perform on OBs vary based on where the OB is being visualized. If the OB is in schedule, as in Figure 21, various actions are possible. By clicking on the lock on the left, the OB becomes fixed and cannot be substituted. The actions on the right instead are:

- Quick view (open eye), will show the main relevant data about the OB.
- Detailed view (magnifying glass) will show every piece of information that is locally stored in the database.
- Visibility check (sketched plot) will show information about the observability of the object, and its ESO probabilities for the current night.
- Substitute (double arrows), will suggest other targets from the database to substitute the target.

Figure 21 - OB visualized when inside schedule.

If instead one is visualizing the OB from the database (either, observable OBs tab or Full OB list window) more actions can be selected (Figure 22):

- Visibility check (as above);
- Quick view (as above);
- Detailed view (as above);
- View in Marshall: a link to the Marshall page for that OB;
- Insert OB into current VES (for manual scheduling, see below Section 7.5.9).

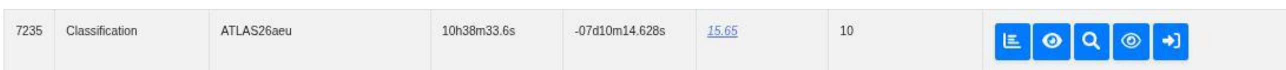


Figure 22 - OB as visualized in observable obs or full OB list.

7.5.2 Actions on schedule

Once the night is opened by the Science team, a schedule is proposed and displayed on the main page (as in Figure 23). The user can work on OBs as shown in 7.5.1 (first part). At the end of the night, a proper button is used to stop operations and close the observing night.

Schedule CloseNight

ID	Scheduled	Target	Type	Ra.	Dec	Obs. Start	Obs. End	Actions
14201	S	GD71				2026-03-19 00:15:17.239	2026-03-19 00:25:17.239	
7601	P	SN2026ns	FOLLOW UP			2026-03-19 00:25:17.446	2026-03-19 01:35:17.446	
7670	P	4FGLJ0649.5-3139	FOLLOW UP			2026-03-19 01:30:17.446	2026-03-19 02:20:17.446	
7794	P	SN2026auh	FOLLOW UP			2026-03-19 02:15:17.446	2026-03-19 03:25:17.446	
8056	G	AT2026cie	CLASSIFICATION			2026-03-19 03:25:17.239	2026-03-19 04:35:17.239	
7669	P	4FGLJ1300.0+1753	FOLLOW UP			2026-03-19 04:40:17.446	2026-03-19 05:55:17.446	
7756	P	AT2026bgd	FOLLOW UP			2026-03-19 05:50:17.446	2026-03-19 06:45:17.446	
7094	P	SwiftJ1357.2-0933	FOLLOW UP			2026-03-19 06:40:17.446	2026-03-19 07:55:17.446	
7093	P	SwiftJ1357.2-0933	FOLLOW UP			2026-03-19 07:50:17.446	2026-03-19 08:45:17.446	
ID	Scheduled	Target	Type	Ra.	Dec	Obs. Start	Obs. End	Actions

Figure 23 - Schedule as visualized in the main page of the SOXS user interface.

7.5.3 OB substitution

Once the schedule has been proposed, each OB can be inspected by the Science Team as in Figure 24.

The schedule can still be modified as it is not synchronized with the P2 database at this point. This can be done by the Science team directly on the Web interface, as already mentioned: they can click on the rightmost blue button of Figure 24 under the "action" category. It will present a two-arrow symbol and it will ask the scheduler to suggest another OB (or group of OBs) to substitute the selected one.

ID	Scheduled	Target	P2 Synched	Obs. Start	Obs. End	Actions
7188		AT2022nlr	YES	2022-11-03 00:30:54.978	2022-11-03 01:35:54.978	

Figure 24 - A snapshot of the schedule as it is viewed by the Science Team.

The scheduler will produce a fake gap in the schedule in place of the current OB, and the GGF will run over the gap for a number of times (currently set to be 5). It will then return the single OB, or group of OBs that fill the gap in an overlaid window as shown in Figure 25. The user can select the OB with the dot on the left, and click select to substitute. If the OB is a Follow-Up target, a warning will be shown before substitution.

OBs Suggested to substitude selected OB (7751)

	ID	Scheduled	Target	Obs. Starts	Obs. Ends	Coordinates	Fixed_time	Slots
	8066	G	AT2026cix	2026-04-11 06:13:19.784	2026-04-11 06:23:19.784	[190.076,-44.8416]	false	[77,78]

Close

Figure 25 - The window that shows possible alternatives to the selected OB.

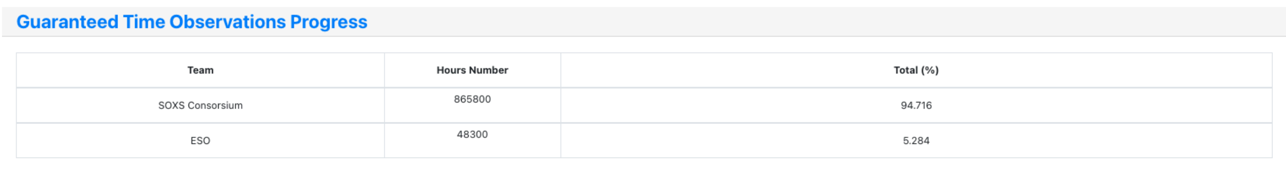
As a further control, the PI or the scientist can select the lock on the leftmost part of the schedule panel(see Figure 24), which will blur the substitution button and forbid the OB to be touched. An unlock is always possible to restore its status.

7.5.4 Night report

In this window, the user (both operator and scientist) can read and/or update reports for the current night. It is also possible to read and download the reports of the previous night.

7.5.5 GTO progress

This window is used to keep track of the GTO. It shows the separation between the SOXS Consortium and ESO, with the relative hours of observation claimed so far, and its percentage (see Figure 26). Data shown here refers to the activities during the commissioning.



Guaranteed Time Observations Progress

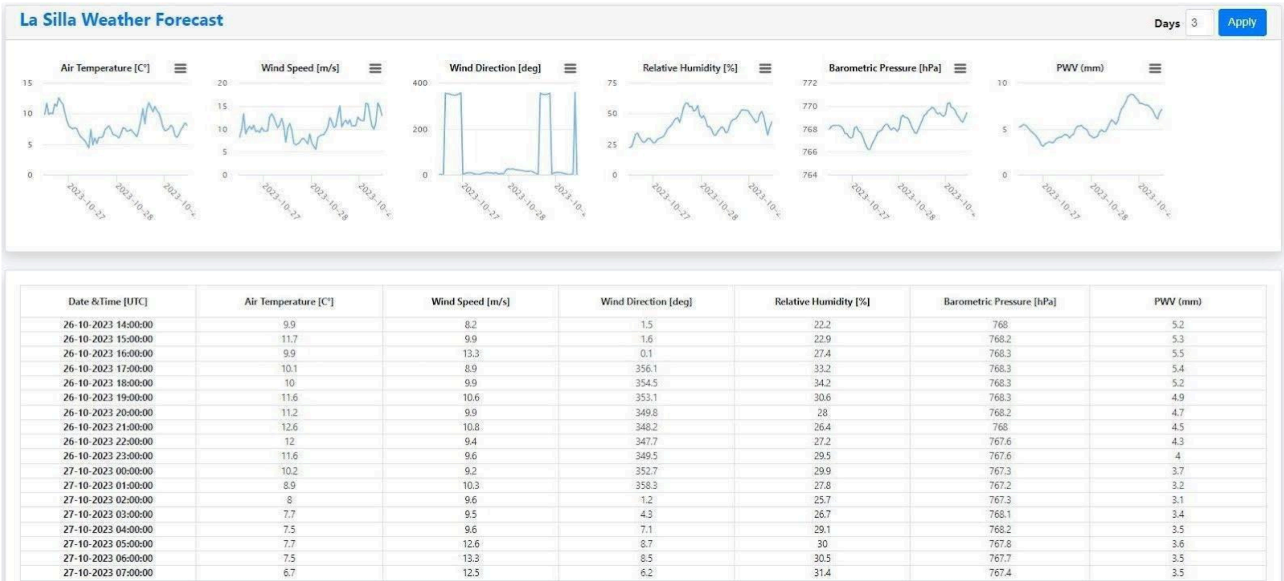
Team	Hours Number	Total (%)
SOXS Consorsium	865800	94.716
ESO	48300	5.284

Figure 26 - GTO progress tracker.

7.5.6 Weather forecasts

The weather forecast window is linked to the weather API in La Silla. The grey bar above is always showing in every part of the interface and reports on current weather conditions. The forecasting below (Figure 27) can be customized by changing the time span in the upper right corner. We currently show:

- Air temperature
- Wind speed
- Wind direction
- Relative humidity
- Barometric pressure
- Precipitable water vapour (PWV)



Date &Time [UTC]	Air Temperature [C°]	Wind Speed [m/s]	Wind Direction [deg]	Relative Humidity [%]	Barometric Pressure [hPa]	PWV (mm)
26-10-2023 14:00:00	9.9	8.2	1.5	22.2	768	5.2
26-10-2023 15:00:00	11.7	9.9	1.6	22.9	768.2	5.3
26-10-2023 16:00:00	9.9	13.3	0.1	27.4	768.3	5.5
26-10-2023 17:00:00	10.1	8.9	356.1	33.2	768.3	5.4
26-10-2023 18:00:00	10	9.9	354.5	34.2	768.3	5.2
26-10-2023 19:00:00	11.6	10.6	353.1	30.6	768.3	4.9
26-10-2023 20:00:00	11.2	9.9	349.8	28	768.2	4.7
26-10-2023 21:00:00	12.6	10.8	348.2	26.4	768	4.5
26-10-2023 22:00:00	12	9.4	347.7	27.2	767.6	4.3
26-10-2023 23:00:00	11.6	9.6	349.5	29.5	767.6	4
27-10-2023 00:00:00	10.2	9.2	352.7	29.9	767.3	3.7
27-10-2023 01:00:00	8.9	10.3	358.3	27.8	767.2	3.2
27-10-2023 02:00:00	8	9.6	1.2	25.7	767.3	3.1
27-10-2023 03:00:00	7.7	9.5	4.3	26.7	768.1	3.4
27-10-2023 04:00:00	7.5	9.6	7.1	29.1	768.2	3.5
27-10-2023 05:00:00	7.7	12.6	8.7	30	767.8	3.6
27-10-2023 06:00:00	7.5	13.3	8.5	30.5	767.7	3.5
27-10-2023 07:00:00	6.7	12.5	6.2	31.4	767.4	3.5

Figure 27 - Weather forecast panel.

7.5.7 Average conditions

This window (Figure 28) shows the registered average observing conditions during observations, which are saved in the database after the completion of observations of an OB. In particular, we take into account a separation between OBs from the Consortium and the ESO community, so that the fairness of the time shared will be ensured.

Figure 28 - Average conditions panel.

7.5.8 Search OB

This overlaid window (Figure 29) allows searching for an OB, or for a category of OBs. By switching the logical operator OR/AND, the algorithm will look either for objects that present one parameter or the others, or it will search for those that respect all of the requests from the query.
Currently the query can be:

- OB owner: either ESO or SOXS.
- OB_ID, unique number associated with each OB in the local database.
- Target name, a string.
- PI Name, among the SOXS users.

Figure 29 - The window for OB searching.

Once the query is launched, the results are displayed as in Figure 30.

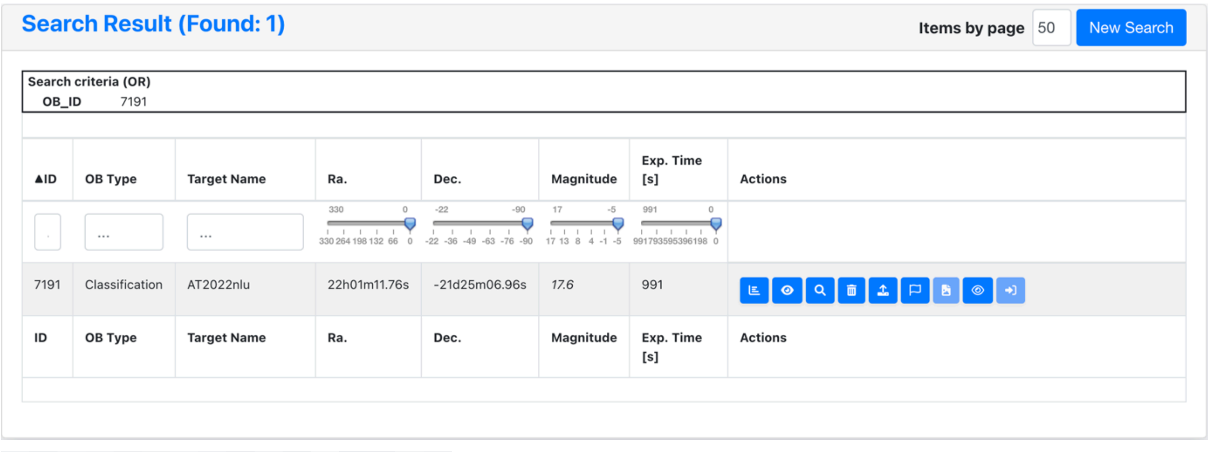


Figure 30 - The window for OB searching result.

7.5.9 Manual Scheduling

Another option for the Science team is to manually build a schedule. However, we do not recommend the use of this feature, since it loses all of the optimization. The Science Team will need to click on the "observable OB" tab, which will show them all the OBs that can be added to the schedule (see Figure 31). The rightmost button in the "action" section will try to add the OB to the schedule. To help the scientist, the "Free Slots" button at the top will return which slots are free to schedule the OBs.

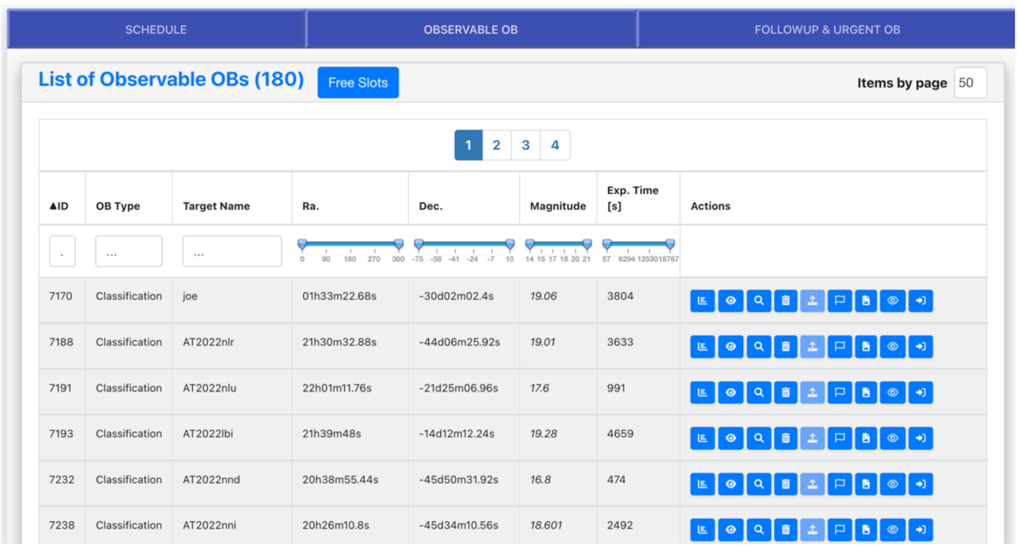


Figure 31 - Observable OBs as shown in the interface.

8 Practical information for the implementation

- The scheduler will be hosted in a physical machine at the INAF-OAB observatory in Merate, where it will continuously run.
- A new schedule will be produced automatically every day. Then, it will be either confirmed or modified during the science team meeting, depending on specific requests by the users, new transients discovered, previously aborted observations to be repeated, etc. The science team meeting will likely be around 16-17 CET (11-13 Chilean time, depending on the period of the year), after which the schedule will be confirmed and "closed". At this point, the schedule will be strictly followed by the TIO during the night unless unexpected circumstances occur, requiring the scheduler to discard it and build a new schedule, as explained in Sect. 6.3.2.
- The approved schedule can be saved and sent to La Silla Observatory before the beginning of the night, so that, should any internet connection issue happen, the TIO will know which OBs to launch during the observing night. In the same way, an "offline" copy of each OB of the approved schedule will be created for the same purpose. Of course, no changes can be made in this particular situation, which we expect to occur extremely rarely. For details, see Sect. 6.3.1.

9 Conclusions and final remarks

The SOXS scheduler commissioning has successfully demonstrated that the system meets its primary design requirements and is ready for regular scientific operations. Through testing during the commissioning runs of August/September 2025 and January 2026, we have validated all critical functionalities of the scheduler, from basic observation planning to complex recovery procedures under adverse conditions.
The major achievements can be summarized as follows:

Robust ESO integration: The scheduler has proven its capability to interact with the ESO Phase 2 system, including synchronization with the P2 database, OB verification, and automated population of the VES. The interaction with the VOT and BOB has been tested extensively and operates reliably, with the only minor limitation being the ~5-minute refresh interval of the VOT, which is inherent to the ESO infrastructure.

Automated night management: The daemons managing the nights have successfully handled various scenarios during real observing nights, including:

- Regular operations following pre-approved schedules
- Recovery from delays and advances in the observation timeline
- Handling of Failed and Aborted observations
- Dynamic response to changing weather conditions
- End-of-Night optimization including standard star observations

The recovery loops have proven to be solid in maintaining high-priority observations while adapting to unforeseen circumstances. This demonstrates how the scheduler is able to maximise the scientific output even under challenging conditions.

Operational Validation: The TIOs, after being instructed, were able to completely manage the upcoming observations, still under our supervision. This test confirmed that the system is intuitive and practical for regular use.

Target diversity: The scheduler has successfully handled a wide range of observational requirements, including fixed and moving targets, urgent target of opportunity and standard star calibrations at various points during the night. Furthermore, SOXS templates have been tested and validated, covering acquisition, science observations and calibration procedures.

System reliability: The commissioning period has demonstrated that the scheduler's core architecture is reliable. The MySQL database structure effectively manages all necessary information, from target catalogs to observing conditions and GTO time tracking. The filtering routines through astroplan provide accurate observability predictions. The two scheduling algorithms (priority and GGF) successfully optimize the telescope utilization.

Readiness: The implemented contingency plan for internet connection failures, though expected to be rarely needed, demonstrates thorough preparation for all eventualities. The ability to provide offline copies of approved schedules and OBs ensures that observations can continue even under exceptional circumstances.

10 Appendix

Below we report the tables present in the local database, with columns and default values.

Table **ob_list**:

- OB_ID INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
- transientBucketID INTEGER DEFAULT 0,
- observing_description_name VARCHAR(200) DEFAULT NULL,
- observer_comment VARCHAR(200) DEFAULT NULL,
- target_name VARCHAR(100) NOT NULL,
- magnitude_list JSON,
- magnitude FLOAT,
- _filter VARCHAR(30),
- right_ascension FLOAT NOT NULL,
- declination FLOAT NOT NULL,
- constraints_name VARCHAR(100),
- airmass FLOAT DEFAULT 2.91,
- sky_transparency INTEGER DEFAULT 2,
- lunar_illumination FLOAT DEFAULT 1.0,
- pwv FLOAT DEFAULT 30.0,
- moon_angular_distance FLOAT DEFAULT 30.0,
- image_quality_arcsec FLOAT DEFAULT 1.5,
- finding_chart_marshall VARCHAR(400),
- finding_chart_eso BOOLEAN DEFAULT 0,
- Auto_OB BOOLEAN DEFAULT 0,
- Follow_up_OB BOOLEAN DEFAULT 0,
- ESO_OB BOOLEAN DEFAULT 0,
- Urgent_OB tinyint(1),
- ESO_Program_ID INTEGER,
- ESO_Run_ID INTEGER,
- in_observation BOOLEAN DEFAULT FALSE,
- ESO_OB_Status CHAR(1),
- ESO_Parent_Container_ID INTEGER,
- Is_Sync_to_P2 BOOLEAN DEFAULT 0,
- ESO_OB_ID INTEGER,
- ESO_User_Priority INTEGER,
- Last_P2_SYNC TIMESTAMP,
- create_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
- last_update_time TIMESTAMP,

- ESO_acqtpl_name VARCHAR(100) default 'SOXS_slt_acq',
- Details_acqtpl JSON,
- ESO_acqtpl_id INTEGER,
- ESO_sctpl_id JSON,
- ESO_sctpl_name VARCHAR(200) DEFAULT 'SOXS_slt_obs_Stare',
- Details_sctpl JSON,
- date_constraint datetime,
- beginning_of_obs_UTC datetime default NULL,
- end_of_obs_UTC datetime default NULL,
- PI_name VARCHAR(500),
- history_magnitude longtext,
- seeing FLOAT DEFAULT 1.5,
- processing_date VARCHAR(500),
- warnings LONGTEXT,
- deltaMagPerJD JSON,
- parent_ob INTEGER DEFAULT NULL,
- fixed BOOLEAN DEFAULT NULL,
- FC_ids JSON,
- concatenated_ob VARCHAR(255),
- Std_OB BOOLEAN DEFAULT NULL,
- db_priority VARCHAR(255),
- moving BOOLEAN DEFAULT FALSE,
- proper_motion_ra VARCHAR(255),
- proper_motion_dec VARCHAR(255),
- adaptive_slit BOOLEAN DEFAULT FALSE,
- start_GTO_time datetime,
- end_GTO_time datetime,
- eph_ids JSON

Table **obs_bucket**:
- OB_ID INTEGER NOT NULL,
- ob_type VARCHAR(20),
- N1 JSON,
- N2 JSON,
- N3 JSON,
- N4 JSON,
- N5 JSON,
- todays_rank VARCHAR(255),
- is_in_VES BOOLEAN DEFAULT FALSE

Table **schedule**:

- night_date VARCHAR(250),
- scheduled_obs JSON,
- status INTEGER DEFAULT 0,
- to_be_synched BOOLEAN DEFAULT FALSE,
- std_check BOOLEAN DEFAULT FALSE

Table **history_schedule**:

- night_date VARCHAR(250),
- scheduled_obs JSON,
- create_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP

Table **nights**:

- id int NOT NULL
- create_time datetime DEFAULT CURRENT_TIMESTAMP,
- update_time datetime DEFAULT NULL,
- night varchar(255) UNIQUE DEFAULT NULL,
- night_closed tinyint(1) NOT NULL DEFAULT 0,
- skip_std_beg BOOLEAN DEFAULT FALSE

Table **failed_obs**:

- OB_ID INTEGER NOT NULL,
- night_date varchar(250),
- cause varchar(100)

Table **average_conditions**:

- time_obs TIMESTAMP DEFAULT CURRENT_TIMESTAMP
- seeing FLOAT
- airmass FLOAT
- wind_dir FLOAT,
- wind_speed FLOAT,
- temperature FLOAT,
- pressure FLOAT,
- dew FLOAT,
- ob_owner varchar(20)

Table **manual_weather**:

- entry_time datetime DEFAULT CURRENT_TIMESTAMP,
- sky_transparency varchar(20),

- seeing_lasilla-fwhm FLOAT,
- meteo_lasilla-wind_dir2 FLOAT,
- meteo_lasilla-wind_speed2 FLOAT,
- meteo_lasilla-temperature2 FLOAT,
- meteo_lasilla-pressure2 FLOAT,
- meteo_lasilla-dew_point2 FLOAT,
- meteo_lasilla-rhum2 FLOAT,

Table **GTO**:
- ob_owner VARCHAR(255) PRIMARY KEY,
- used_time FLOAT DEFAULT 0

Table **Report**:
- night_date DATE NOT NULL,
- user VARCHAR(50),
- time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
- report JSON,
- entry_number int

Table **soxs_USERS**:
- user_name VARCHAR(300) NOT NULL,
- email VARCHAR(300),
- phone VARCHAR(300))

Table **daemon**:
- running_status BOOLEAN default FALSE

Abbreviations and Acronyms

AD	Applicable Document
API	Application Program Interface
DICD	Data Interface Control Document
EFOSC2	ESO Faint Object Spectrograph and Camera
ESO	European Southern Observatory
ETC	Exposure Time Calculator
FITS	Flexible Image Transport System
FRB	Fast Radio Burst
GRB	Gamma-Ray Burst
GTO	Guarantee Time Observation

GW	Gravitational Wave
HC	Health Check
INAF	Istituto Nazionale di AstroFisica
INS	Instrument Software
LSST	Large Synopthic Survey Telescope
MoU	Memorandum of Understanding
NTT	New Technology Telescope
OB	Observing Block
OPC	Observing Programmes Committee
P2PP	Phase 2 Proposal Preparation
PSP	Public Survey Panel
PI	Principal Investigator
QC	Quality Control
REM	Rapid Eye Mount
SDP	Science Data Product
SOXS	Son Of X-Shooter
TBD	To Be Defined
ToO	Target of Opportunity
VLT	Very Large Telescope
WP	Work Package