TSO

Progress Report 1

Sebastian Lopez
Gustavo Andres Murcia
Kirk Vander Ploeg
Elijah Ward

UWO CS 4470Y November 5, 2018

T | *Table of Contents*

1 | *Project Description*

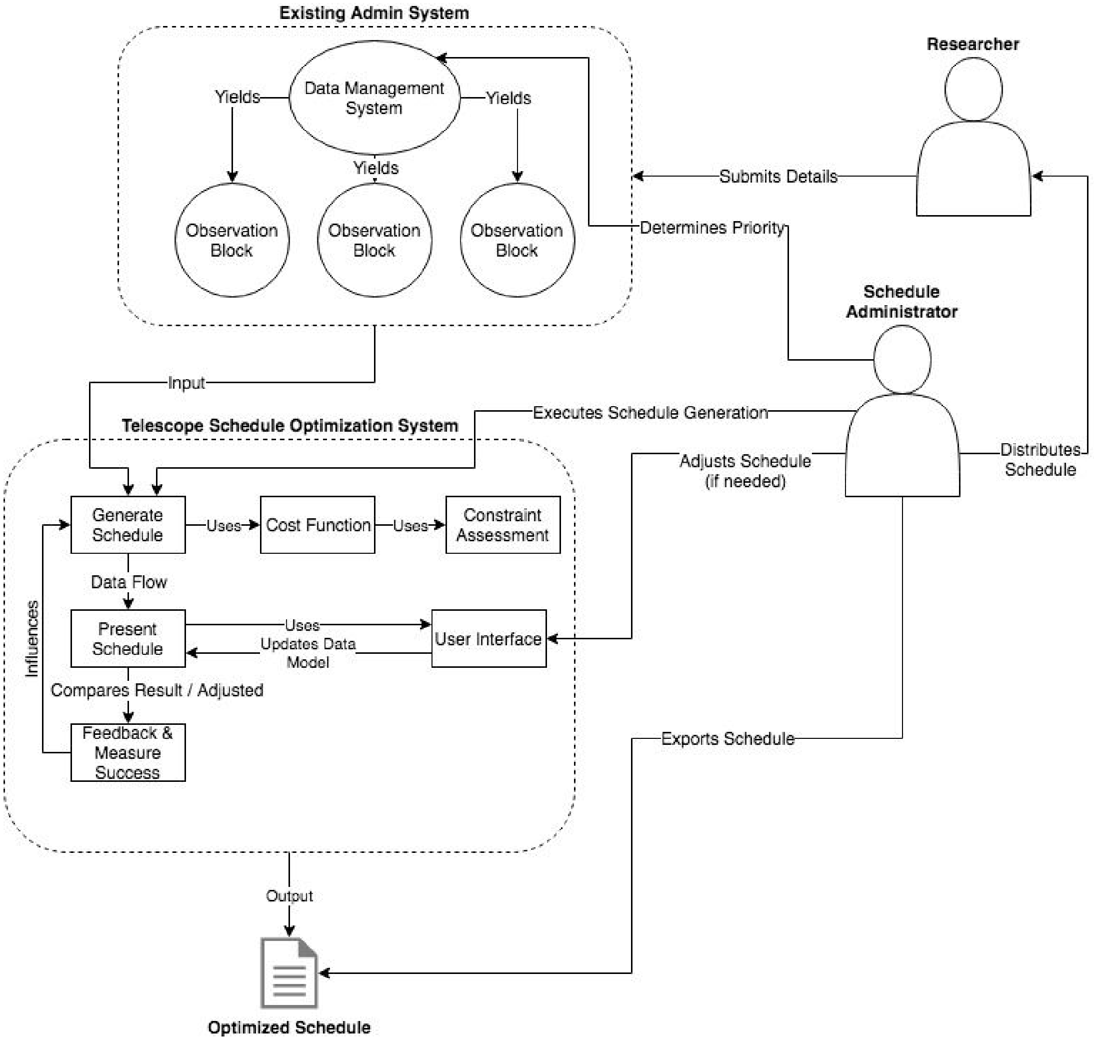
Summary

The following is the first of four reports on the progress of our chosen capstone project: ***Telescope Scheduling Optimization***. This project's goal is to develop a Telescope Scheduling Optimizer system for the Maunakea Spectroscopic Explorer (MSE) herein referred to as **TSO**. The members of our team are Elijah Ward, Gustavo Murcia, Kirk Vander Ploeg and Sebastian Lopez. This project is being supervised by Professor Pauline Barmby. Following discussions with Pauline, we have been able to define a high-level understanding of an existing system, expectations of users, and scope of work to be completed within this project. Furthermore, it will specify the roles that have been assigned to each team member for the duration of the project and outline a high-level plan for the target delivery of each component of the project.

Problem Statement

The Canada-France-Hawaii Telescope (CFHT) team is currently operating using a suboptimal method of producing an observation schedule for researchers that wish to use their telescope for their work. The current method is a very manual and labour-intensive process. Administrators must collect all applications, assess the priority of each of the studies associated with the request, consider a set of **fixed** and **dynamic** constraints and do their best to create a schedule that is optimal. This approach is prone to human error and subjectivity. Before the construction of the newly proposed telescope, the MSE, the team desires a more automated way to create an optimized schedule based on the priority, fixed and dynamic constraints of each observation request.

Context Diagram



Stakeholders

Professor Pauline Barmby

Our key stakeholder to whom this project will be delivered is Professor Pauline Barmby. She will also be acting as the project's supervisor. Pauline is an accomplished researcher and instructor within the faculty of Physics and Astronomy at the University of Western Ontario. Her research specializes in observational studies of nearby galaxies and their star clusters. She has made hundreds of academic contributions in her field of expertise, many of which are directly related to observing celestial bodies with different types of telescopes. Her domain knowledge and experience will be paramount to the success of this project.

Maunakea Spectroscopic Explorer Team

The end users for the TSO will ultimately be members of the MSE administrative team who create optimal schedules based on the information that they receive. The current process is heavily manual and could benefit greatly from both time and cost savings that an automated solution would provide to their team, which depends largely on grants for funding. By extension, the researchers who wish to use the MSE telescope to complete essential observations for their research will benefit as well.

It is our hope that the automated solution we deliver will provide schedules that are more considerate and predictive of environmental factors that often detract from a quality observation. The team has discussed including this as a key performance indicator if we can obtain enough data about the performance of the existing solution.

Background

As of the start of this TSO system, the MSE telescope has not begun construction and is awaiting funding. The plan is to house the MSE within the enclosure of the CFHT which has been operational since 1979. The MSE will extend the telescope's observational capabilities to a more modern standard. As a result, all of the information provided to our team about the current state of their scheduling capabilities are coming from the CFHT team.

Existing Solution

Following preliminary discussions with Professor Pauline Barmby, it is the team's understanding that the software currently in use is of a different class to the system we aim to deliver. The tool currently being used by the CFHT team is much more of an **organizational aid** than a **scheduler**. Researchers submit requests to use the telescope to gather observations related to a specific body of research. The Schedule Administrators (SA) consider a multitude of factors to determine which projects will be granted observation time and what the optimal schedule should be given the nature and priority of each of the projects. The SA uses the current tool as an aid to aggregate data that is relevant to forming a schedule and then proceeds to manually decide what the optimal schedule is.

Proposed Solution

Technologies and Tools

Following discussions with Professor Barmby, the team has decided to implement the proposed system as a portable solution written in python. There is an existing ecosystem of libraries that have been developed for the purpose of performing calculations related to astronomical observation scheduling. Given that the scheduling algorithm itself will need to be extremely performant, we may need to write parts of the system in C/C++. We will also need to be sure that any libraries we plan on adopting are using C/C++ bindings for computationally intensive components. Professor Barmby specifically referred us to several existing libraries that will assist us with calculations that would require a deep understanding of physics and astronomy.

Python

As a high-level programming language that is known for its ease of use when prototyping a solution involving scientific computation, we feel that python is the perfect choice for this project. There are a wealth of libraries and tools that help teams to create solutions much like the one that we propose. In recent years Python has matured greatly not only within the scientific community but also within production systems of leading organizations. Between the immense support from the open source community and the fact that industry leaders are using python to deliver massively scaled systems, we have confidence that a solution we create in this language can continue to be useful for our stakeholders into the future.

Astropy

Astropy is an open source python package recommended to us by Professor Barmby. It is created and maintained by experts in the fields of astronomy and physics. With over 22,000 commits, 1800 stars and 269 contributors on GitHub, we look forward to further investigating the use of what could potentially be an extremely integral part of our proposed system. Astropy also specifically provides functionality for scheduling observations with their Astroplan module. We anticipate that this module can help us by being part of the input for our cost function to identify the most optimal schedule. For more information about Astropy look to http://www.astropy.org/.

Theory and Methodology

In very simple terms, what we are trying to achieve is to output the best possible schedule taking into account what celestial bodies will be observed and all the necessary data for said observation. Our current approach will deal with dividing each schedule into manageable units known as observation blocks. These Blocks will then be grouped together to create a schedule which will be executed and evaluated through the formulating of a **cost function** that will evaluate the candidate schedule's effectiveness.

Each observation block will have a number of constraints that will influence its placement within a schedule. So far each observation block consists of the following constraints: Fixed; dictates whether the observation block will be used for a long or short period of time in order to gather all the data necessary. Dynamic; unpredictable conditions such as the weather will definitely be a limitation that will need to be accounted for. Priority; the observation of certain celestial bodies may be more important than others.

A schedule will then be produced as a combination of observation blocks, and then our job will be to measure the effectiveness of each respective schedule. Currently, the metrics used to measure the effectiveness of a schedule are completion percentage and quality of observation during each observation block. During a given observation block it may not be possible to observe the target body for the entire period of time due to dynamic conditions such as the weather. We should be able to measure the success of our generated schedules using some predefined metrics. We can build a "completion" metric by assessing the progress made towards the completion of all necessary observations for the project during that observation block, reported by the observing researchers. We can also discern a "quality" metric for each observation block by assessing the percentage of achieved progress within the duration of the block, relative to the total estimated time to complete the observation. From these metrics we can qualify the success or shortcomings of our chosen approach, giving the opportunity to those who wish to improve it in the future.

With respect to the more technical aspect of our methodology, we are currently looking into using **genetic algorithms** as an optimization method. Through the use of **genetic algorithms**, we hope to find the best configuration of observation blocks considering their respective constraints by generating a population of schedules and iterating to minimize a cost function. Ideally, we will have access to historical data in order to test our approach against real conditions. This cost function, which will be evaluated for each candidate schedule, will likely consider inputs such as historical metrics. It will also need to include input about dynamic constraints such as weather data, and fixed constraints such as observation window of a celestial body and priority for use of telescope resources.

Team Member Roles

The team has arrived at an assignment of roles based on a discussion of personal interests and experience. In addition, the team has come to a set of mutual agreements in order to ensure that our workflow during the execution of the project is as smooth as possible. The first is that every team member will serve as a backup to the Project Manager in order to ensure accountability to our deliverables and maintain an open conversation about what we expect from ourselves and each other. This also ensures that any team member will be prepared to assume project management responsibilities in the absence of the Lead Project Manager. The second is that every team member will be responsible for supporting the Quality Assurance Lead by creating unit tests for their own work which will be subject to review by the QA lead on completion.

Role Assignment Table

Team Member	Primary Role(s)	Secondary Role(s)
Sebastian Lopez	- Lead Requirements Analyst - Lead Tools Specialist	- Backup Architect
Gustavo Murcia	- Lead UX Design - Lead Project Manager	- Documentation Lead - Backup Solutions & Requirements Analyst
Kirk Vander Ploeg	- Quality Assurance Lead - Integration Testing	- Backup UX Design
Elijah Ward	- Lead Architect	- Backup Tools Specialist - Backup Project Manager

3 | *System Requirements*

The following is a collection of requirements that the TSO system will adhere to. Through communications with Pauline, consultation of the existing solution, and our plans for TSO, we have captured these requirements and categorized each one as either a Functional Requirement (FR) or a Quality Requirement (QR). When requirements need modification, they are to be verified by both Pauline and the team.

System Requirements *Found in Appendix*

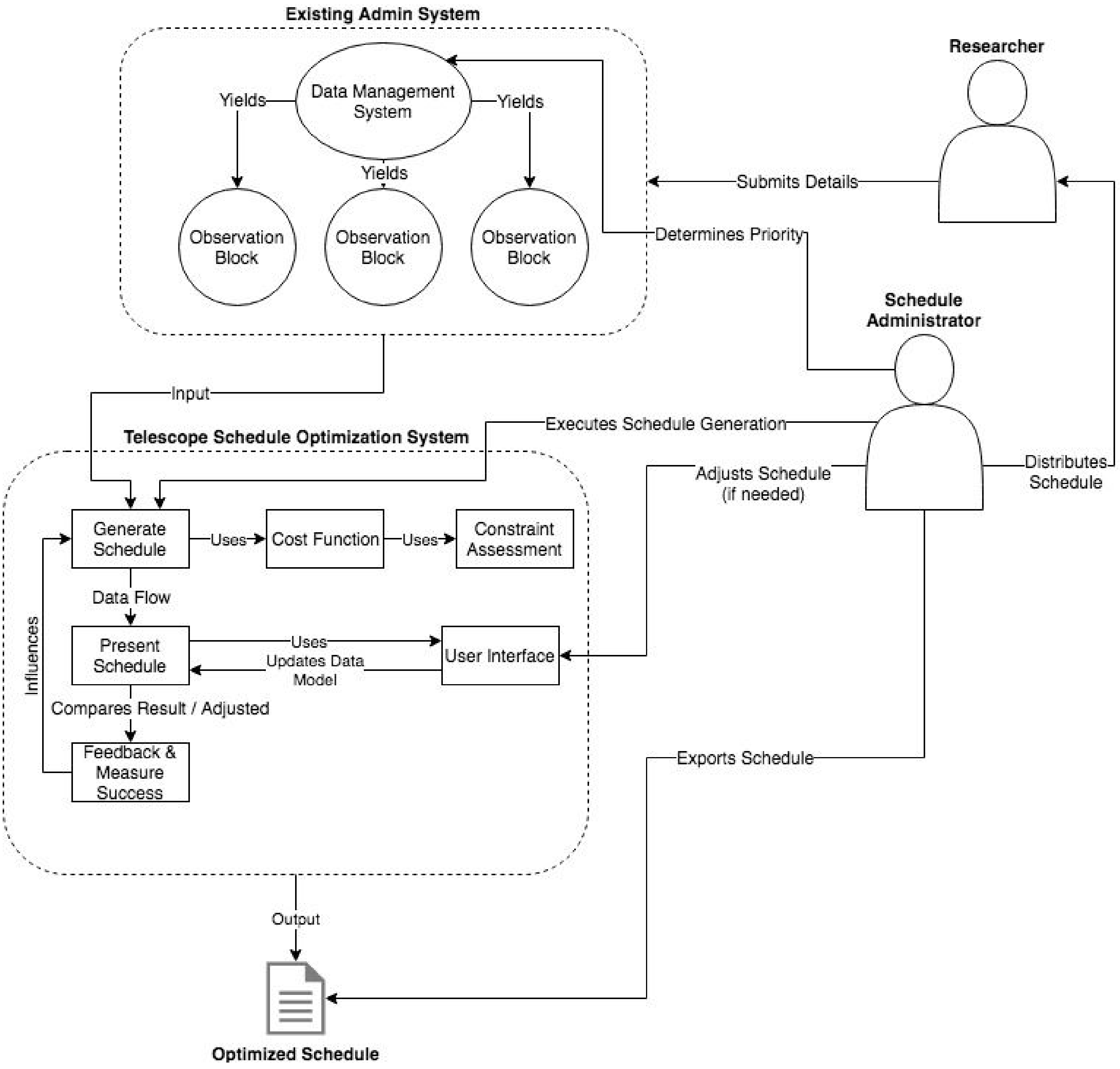
4 | *Project Plan*

To best manage our team of 4 along with improving our success with our stakeholders, the following documents are provided. We will be hosting these documents on the Google Cloud in order to improve the efficiency of our own team's internal progress as well as the communication we experience with Pauline and our stakeholders.

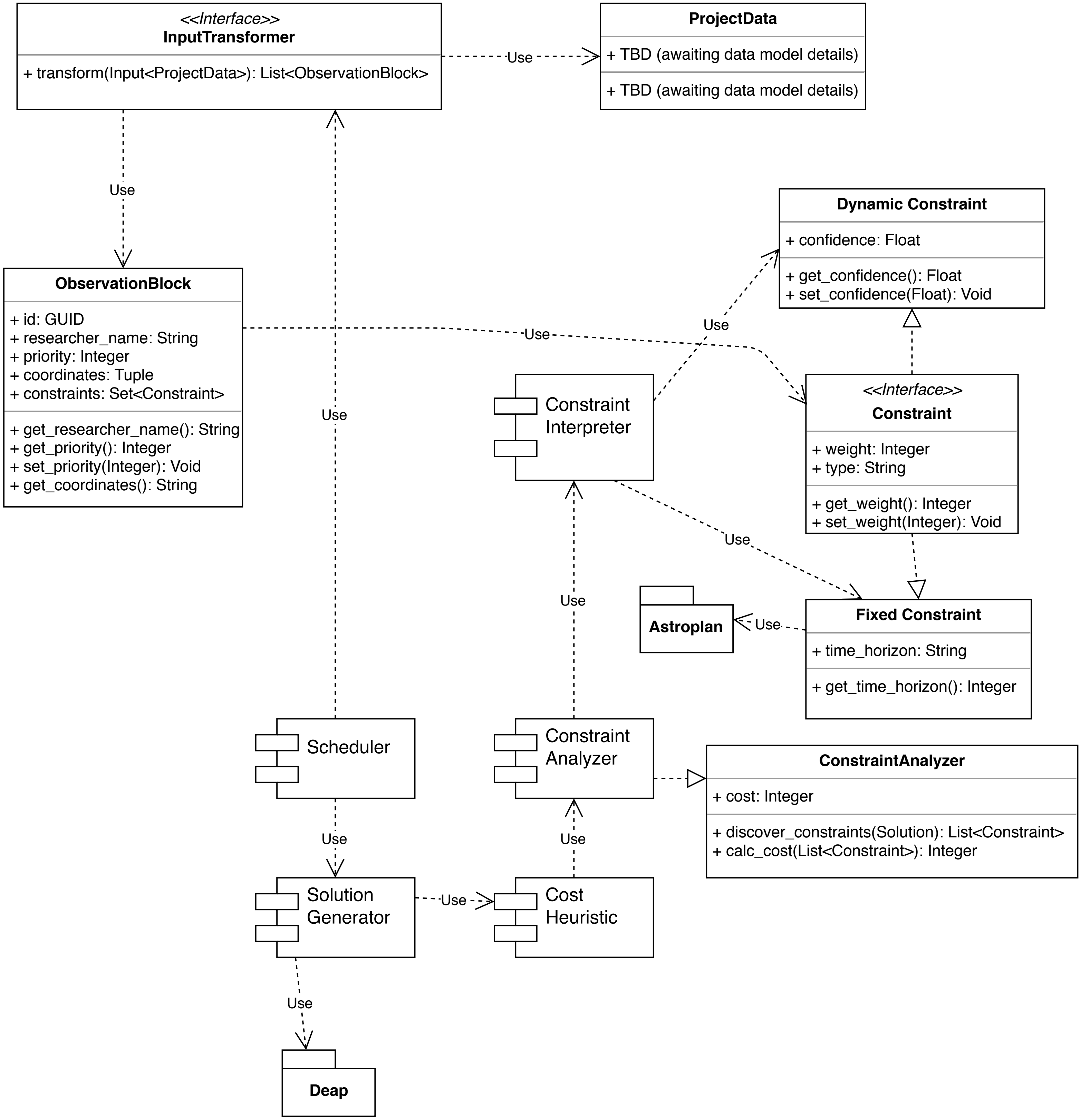
Project Tasks	*Found in Appendix*
Project Timeline	*Found in Appendix*
Project Work Breakdown Structure	*Found in Appendix*

A | *Appendix*

Context Diagram



TSO High-Level System Architecture



MSE:TSO - Requirements

Type	ID	Requirement Title	Status	Category	Requirement	Detail	Related	Notes	Client Notes
Feature	1	Data Import	Review	Import, UI	*These are the requirements around the TSOs capability of importing data from external sources.*	The TSO System will require data to be imported for the following types of data. Program Data and Constraint Data			
FR	1.1	Program Data Import	Review	Import	TSO's main purpose is to act upon Program data. This data represents the various "programs" that are to be optimally scheduled by the TSO. This will have to be imported into the TSO system from various sources.			Format and Nature of data to be determined. How many import sources?	
QR	1.1.2	Program Data Error Checking	Review	Errors	TSO will ensure that any incoming data will not create data bugs because of the nature of the incoming data.	An example of this could be program data that contains unsupported characters			
FR	1.2	Constraint Data Import	Review	Import	In order to schedule Program data, TSO requires data that will determine whether some programs are fit to be scheduled or not.	An example of a constraint is the weather. A Program X could rely on a condition that the atmospheric condition of the night is of a certain value.		Format and Nature of data to be determined. How many import sources?	
Feature	2	Data Export	Review	Export, UI	*These are the requirements around the TSOs capability of exporting data it processes.*	Key functionality of the TSO system, that allows users to use their results outside of the system itself. Schedules can be exported in a variety of formats, while the data used to create these will also be made available.			
FR	2.1	Data Export Action	Review	Export	A user will request TSO to output the final schedule to the desired output format.	Gives the user the freedom to use and view the schedule outside of the TSO.			
FR	2.1.1	Data Export Format	Review	Export	A user can export data from TSO as a database (SQL, Mongo...)			Confirm Data Export Format	
FR	2.1.2	Data Export Format	Review	Export	A user can export data from TSO as an Excel file			Confirm Data Export Format	
FR	2.1.3	Data Export Format	Review	Export	A user can export data from TSO as a PDF file			Confirm Data Export Format	
FR	2.2	Final Data Visualization	Review	UI	The TSO system will allow a user to view the final schedule the system has determined to be optimal.				
Feature	3	Observation Data	Review	Data, UI	*These are the requirements around the TSOs capability to represent observation data to be processed by the scheduler component of the TSO.*				
FR	3.1	Data Identification	Review	Data	The user shall be able to identify the source of the data they have imported or manually entered into TSO	By uniquely identifying the data in the system, no collisions will occur and correct output will be produced			
FR	3.2	Data Model	Review	Data	The TSO system will manage it's own data model. It will closely resemble the data model of the import, but could have changes to adapt to the scheduling process used.				
FR	3.2.1	Data View	Review	Data, UI	The user shall be able to view the data that TSO is able to schedule	A UI that displays a common format for Observation Data could be something that will improve the process of selecting data to be scheduled		Does the current CFHT system do this? Is this something they want?	
FR	3.2.2	Data Creation	Review	Data	The user shall be able to add data to the TSO system	This is covered through the Data Import requirements	1.*		
FR	3.3	Data Persistence	Review	Data	The TSO system shall persist data between sessions of use	This could either be through a shared database or individual local files on the machines running the software.		Super important one to know now. This dictates 7.* too and honestly, it might be scope creep	
Feature	4	Scheduler	Review	Scheduling, UI, Data	*These are the requirements around the TSOs capability to schedule observation data.*	The TSO should produce an optimal schedule taking into account the proposed blocks and the best way to organize these. The output should be presented in a clear, detailed and legible manner.			
FR	4.1	Scheduling Selection	Review	Scheduling, UI, Data	A user shall be able to select which observation blocks to consider for the current scheduling		3.*		
FR	4.2	Genetic Algorithm	Review	Scheduling	The TSO will use an iterative proces of a genetic algorithm using a cost function to determine the best outcome for the final schedule from a selection of N observation blocks	This is a macro level event. This is how the TSO will find the optimal solution		Currently a High Level. Further design is required to determine additonal FRs and QRs	
QR	4.2.2	Genetic Algorithm Performance	Review	Quality	The TSO's use of genetic algorithms will be tuned to meet speed and quality KPIs set by stakeholders	If the genetic algorithm is tunable, it shall be tunable to paramters that are concrete to the genetic algorithm. For example: One such KPI could be "redundancy" - "As schedulers, we would like to see the next best schedule"			
FR	4.2.1	Cost Function	Review	Scheduling	The TSO will utilize a custom cost function to determine the final schedule from a set of N observation blocks	This is a micro level event. This is how the TSO will find the best solution given a set of constraints		Currently a High Level. Further design is required to determine additonal FRs and QRs	
Feature	5	Schedule Analysis	Review		*These are the requirements around the TSOs capability to analyze the schedules it produces.*	In order to produce the most optimal schedules, the TSO system lets the user choose schedules to analyze and provide an adequate feedback report stating the effectiveness of the schedule.		These Requirements will be filled out when we know exactly what properties of a produced schedule will be analyzed or not. This Req family could easily become something we develop internally	
Feature	6	Roles	Review	Users, UI	*These are the requirements around the TSOs capability to have different users perform different actions.*	The TSO system will have seperate interfaces for those users seeking to submit a observation block proposal, and a seperate one for the management of proposals, creation and analysis of schedules.			
FR	6.1	TSO User	Review	Users, UI	The TSO User will use the system to integrate previously developed systems of data and telescope control into their work stream.	The TSO systems is meant to be used by a small group of people working at the current CFHT site. Because of this, there will not be a need to introduce the idea of a multi-user multi-role system.			
Feature	7	Performance	Review		*These are the requirements around the TSOs performance goals.*				
QR	7.1	Historical Performance	Review	Scheduling, Quality	If provided with historical Data, the TSO will be able to match the accuracy of previous schedules. In this way, we will be able to verify TSO against the performance of the previous system				

MSE:TSO - Project Tasks

FRs and QR have been mapped to line items to work on. The "Task" type will be a child of the given ID plus it's own ID.

Example: Task 1.1.1 is the first task under the FR/QR 1.1

Legend	
Waiting	W
Done	D
In Progress	IP
Blocked	B

Type	ID	Title	Category	Assignee	Notes	Notes+	Status
	M	Deliverables		ALL	*These tasks indicate the 4 Milestones (aka: Deliverables) that we must submit for the course. Each one represents a unit of work towards Code, Documentation, Client Feedback,..etc*		W
Task	M.1	Milestone 1		ALL	Progress Report 1: 15%, due November 5, 2018		D
Task	M.1.1	Overall project Description		ALL			D
Task	M.1.2	Context Diagram		Elijah			D
Task	M.1.3	Role Matrix		Kirk			D
Task	M.1.4	System Requirements		Sebastian			D
Task	M.1.5	Project Plan		Gustavo			D
Task	M.1.6	Preliminary sketch of the system's WBS		Gustavo	OPTIONAL		D
Task	M.2	Milestone 2		ALL	Progress Report 2: 25%, due December 3, 2017		W
Task	M.2	Milestone 3		ALL	Progress Report 3: 20%, due February 25, 2018		W
Task	M.2	Milestone 4		ALL	FINAL Report and presentation: 40%, Report due April 8, 2018		W
Feature	1	Data Import		ALL	*The Data Import component*		W
FR	1.1	Program Data Import		ALL			W
Task	1.1.1	Design Program Data Input Transformer	Design	ALL			W
Task	1.1.2	Develop Program Data Input Transformer	Dev	Gustavo			W
QR	1.1.2	Error Checking Incoming Data	QA	Kirk			W
Task	1.1.3	Develop Associated UI	UI	Kirk			W
FR	1.2	Constraint Data Import		ALL			W
Task	1.2.1	Design Constraint Data Import	Design	ALL			W
Task	1.2.2	Develop Constraint Data Import	Dev	Sebastian			W
Task	1.2.3	Develop Associated UI	UI	Gustavo			W
Feature	2	Data Export		ALL	*The Data Export component*		W
FR	2.1	Data Export Action		ALL			W
Task	2.1.1	Develop Associated UI	UI	Kirk			W
FR	2.1.1	Data Export to Format(s)		ALL			W
Task	2.1.1.2	Design Schedule Consumer for Export(s)	Design	Sebastian			W
Task	2.1.1.2	Develop Schedule Consumer for Export(s)	Dev	Elijah			W
FR	2.2	Final Data Visualization		ALL			W
Task	2.2.1	Develop Associated UI	UI	Gustavo			W
Feature	3	Observation Data		ALL	*The Observation Data component*		W
FR	3.1	Data Identification		ALL			W
Task	3.1.1	Develop interal ID tracker for managed data.	Dev	Elijah			W
FR	3.2	Data Model		ALL			W
Task	3.2.1	Investigate existing data model from CFHT	Investigate	Elijah			W
Task	3.2.2	Design TSO Data Model	Design	ALL			W
Task	3.2.3	Develop TSO Data Model	Dev	Elijah			W
FR	3.2.1	Data View		ALL			W
Task	3.2.1.1	Develop Simplified UI View of TSO Data	UI	Kirk			W
FR	3.3	Data Persistence		ALL			W
Task	3.3.1	Design strategy for persiting user data.	Design	Sebastian			W
Feature	4	Scheduler		ALL	*The Scheduler component*		W
FR	4.1	Scheduling Selection		ALL			W
Task	4.1.1	Design Schedule Structure	Design	Sebastian			W
Task	4.1.2	Develop Schedule Structure	Dev	Kirk			W
Task	4.1.3	Design Associated UI to select TSO Data	UI	Gustavo			W
FR	4.2	Genetic Algorithm		ALL			W
Task	4.2.1	Design Genetic Algorithm	Design	ALL			W
Task	4.2.2	Develop Genetic Algorithm	Dev	Elijah			W
QR	4.2.2	Optimize implementation to improve KPIs	Optimize	Gustavo			W
FR	4.2.1	Cost Function		ALL			W
Task	4.2.1.1	Design Cost Function	Design	ALL			W
Task	4.2.1.2	Develop Cost Function	Dev	Sebastian			W
Feature	5	Schedule Analysis		ALL	*The Schedule Analysis component*		W
Task	5.1	Investigate existing Schedulers to determine optimization process	Investigate	Elijah			W
Feature	6	Roles		ALL	*The Roles component*		W
FR	6.1	TSO User		ALL			W
Task	6.1.1	Develop system executable delivery method	Dev	Sebastian			W
Feature	7	Performance		ALL	*The Performance component*		W
QR	7.1	Historical Performance		ALL			W
Task	7.1.1	Develop integration test of TSO given historical data	Test	Kirk			W

MSE:TSO - Project Timeline

Legend		
Waiting	W	.
Done	D	;
In Progress	IP	-
Blocked	B	x

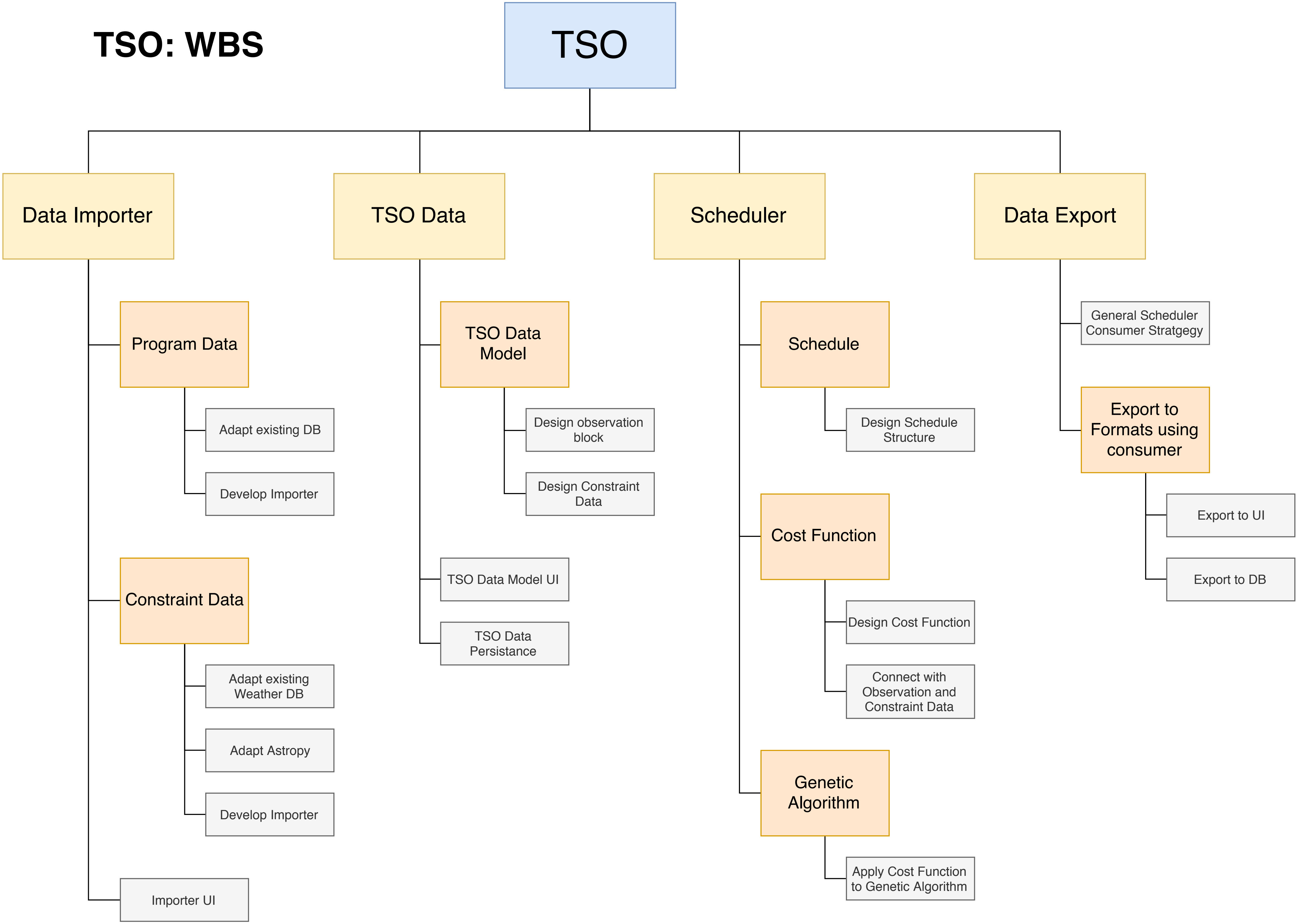
(Column = Week as of Monday)

The status placed in a dark grey header indicates to general timeline of when that high level task is being worked on

CURRENT

	ID	Title	Category	Assignee	Notes	Notes+	Status	Oct-1	Oct-8	Oct-15	Oct-22	Oct-29	Nov-5	Nov-12	Nov-19	Nov-26	Dec-3	Dec-10	Dec-17	Dec-24	Dec-31	Jan-7	Jan-14	Jan-21	Jan-28	Feb-4	Feb-11	Feb-18	Feb-25	Mar-4	Mar-11	Mar-18	Mar-25	Apr-1	Apr-8
	M	Deliverables		ALL	*These tasks indicate the 4 Milestones (aka: Deliverables) that we must submit for the course. Each one represents a unit of work towards Code, Documentation, Client Feedback,..etc*		W			;	;	;	;				.												.						.
Task	M.1	Milestone 1		ALL	Progress Report 1: 15%, due November 5, 2018		D						;																						
Task	M.1.1	Overall project Description		ALL			D			;																									
Task	M.1.2	Context Diagram		Elijah			D				;																								
Task	M.1.3	Role Matrix		Kirk			D			;																									
Task	M.1.4	System Requirements		Sebastian			D			;																									
Task	M.1.5	Project Plan		Gustavo			D			;	;	;																							
Task	M.1.6	Preliminary sketch of the system's WBS		Gustavo	OPTIONAL		D					;																							
Task	M.2	Milestone 2		ALL	Progress Report 2: 25%, due December 3, 2017		W										.																		
Task	M.2	Milestone 3		ALL	Progress Report 3: 20%, due February 25, 2018		W																						.						
Task	M.2	Milestone 4		ALL	FINAL Report and presentation: 40%, Report due April 8, 2018		W																												.
Feature	1	Data Import		ALL	*The Data Import component*		W								.	.	.	.	.	.															
FR	1.1	Program Data Import		ALL			W								.	.	.	.	.	.															
Task	1.1.1	Design Program Data Input Transformer	Design	ALL			W								.	.																			
Task	1.1.2	Develop Program Data Input Transformer	Dev	Gustavo			W									.	.	.																	
QR	1.1.2	Error Checking Incoming Data	QA	Kirk			W										.	.																	
Task	1.1.3	Develop Associated UI	UI	Kirk			W											.	.	.															
FR	1.2	Constraint Data Import		ALL			W								.	.	.	.	.	.															
Task	1.2.1	Design Constraint Data Import	Design	ALL			W								.	.	.																		
Task	1.2.2	Develop Constraint Data Import	Dev	Sebastian			W									.	.	.																	
Task	1.2.3	Develop Associated UI	UI	Gustavo			W											.	.	.															
Feature	2	Data Export		ALL	*The Data Export component*		W																					.	.	.	.	.			
FR	2.1	Data Export Action		ALL			W																								.				
Task	2.1.1	Develop Associated UI	UI	Kirk			W																								.				
FR	2.1.1	Data Export to Format(s)		ALL			W																					.	.	.	.				
Task	2.1.1.2	Design Schedule Consumer for Export(s)	Design	Sebastian			W																					.	.						
Task	2.1.1.2	Develop Schedule Consumer for Export(s)	Dev	Elijah			W																						.	.	.				
FR	2.2	Final Data Visualization		ALL			W																								.	.			
Task	2.2.1	Develop Associated UI	UI	Gustavo			W																								.	.			
Feature	3	Observation Data		ALL	*The Observation Data component*		W						.	.	.	.	.	.			.	.	.	.											
FR	3.1	Data Identification		ALL			W																												
Task	3.1.1	Develop interal ID tracker for managed data.	Dev	Elijah			W																												
FR	3.2	Data Model		ALL			W						.	.	.	.																			
Task	3.2.1	Investigate existing data model from CFHT	Investigate	Elijah			W						.	.																					
Task	3.2.2	Design TSO Data Model	Design	ALL			W							.	.	.																			
Task	3.2.3	Develop TSO Data Model	Dev	Elijah			W										.																		
FR	3.2.1	Data View		ALL			W										.	.																	
Task	3.2.1.1	Develop Simplified UI View of TSO Data	UI	Kirk			W											.																	
FR	3.3	Data Persistence		ALL			W														.	.	.	.											
Task	3.3.1	Design strategy for persiting user data.	Design	Sebastian			W														.	.	.	.											
Feature	4	Scheduler		ALL	*The Scheduler component*		W							.	.	.	.	.	.	.	.	.	.	.	.	.	.								
FR	4.1	Scheduling Selection		ALL			W							.	.	.																			
Task	4.1.1	Design Schedule Structure	Design	Sebastian			W							.	.																				
Task	4.1.2	Develop Schedule Structure	Dev	Kirk			W								.	.																			
Task	4.1.3	Design Associated UI to select TSO Data	UI	Gustavo			W										.																		
FR	4.2	Genetic Algorithm		ALL			W														.	.	.	.	.	.	.								
Task	4.2.1	Design Genetic Algorithm	Design	ALL			W														.	.	.												
Task	4.2.2	Develop Genetic Algorithm	Dev	Elijah			W															.	.	.	.	.									
QR	4.2.2	Optimize implementation to improve KPIs	Optimize	Gustavo			W																		.	.	.								
FR	4.2.1	Cost Function		ALL			W									.	.	.	.	.															
Task	4.2.1.1	Design Cost Function	Design	ALL			W									.	.	.	.																
Task	4.2.1.2	Develop Cost Function	Dev	Sebastian			W											.	.	.															
Feature	5	Schedule Analysis		ALL	*The Schedule Analysis component*		W																				.	.	.	.					
Task	5.1	Investigate existing Schedulers to determine optimization process	Investigate	Elijah			W																				.	.	.	.					
Feature	6	Roles		ALL	*The Roles component*		W																											.	.
FR	6.1	TSO User		ALL			W																											.	.
Task	6.1.1	Develop system executable delivery method	Dev	Sebastian			W																											.	.
Feature	7	Performance		ALL	*The Performance component*		W																								.	.	.		
QR	7.1	Historical Performance		ALL			W																								.	.	.		
Task	7.1.1	Develop integration test of TSO given historical data	Test	Kirk			W																								.	.	.		

TSO: WBS



- TSO
  - Data Importer
    - Program Data
      - Adapt existing DB
      - Develop Importer
    - Constraint Data
      - Adapt existing Weather DB
      - Adapt Astropy
      - Develop Importer
    - Importer UI
  - TSO Data
    - TSO Data Model
      - Design observation block
      - Design Constraint Data
    - TSO Data Model UI
    - TSO Data Persistance
  - Scheduler
    - Schedule
      - Design Schedule Structure
    - Cost Function
      - Design Cost Function
      - Connect with Observation and Constraint Data
    - Genetic Algorithm
      - Apply Cost Function to Genetic Algorithm
  - Data Export
    - General Scheduler Consumer Stratgegy
    - Export to Formats using consumer
      - Export to UI
      - Export to DB