

Observatory Automation Control System  
Shilo Gilor & Amiel Liberman  
Software Design Document

Names: Amiel Liberman & Shilo Gilor  
Date: (08/01/2021)

## TABLE OF CONTENTS

<b>1.</b>	<b>INTRODUCTION</b>	<b>2</b>
1.1	Purpose	2
1.2	Scope	2
1.3	Overview	2
1.4	Definitions and Acronyms	2
<b>2.</b>	<b>SYSTEM OVERVIEW</b>	<b>3</b>
<b>3.</b>	<b>SYSTEM ARCHITECTURE</b>	<b>3</b>
3.1	Architectural Design	3
3.2	Decomposition Description	4
3.3	Design Rationale	4
<b>4.</b>	<b>DATA DESIGN</b>	<b>4</b>
4.1	Data Description	4
4.2	Data Dictionary	5
<b>5.</b>	<b>COMPONENT DESIGN</b>	<b>5</b>
<b>6.</b>	<b>HUMAN INTERFACE DESIGN</b>	<b>5</b>
6.1	Overview of User Interface	5
6.2	Screen Objects and Actions	5

# 1. INTRODUCTION

## 1.1 Purpose

Ariel University is installing an Observatory and would like to automate it as much as possible so we as our final project will help with starting the steps to automate it, we will explain how it is built and how we will connect to the machinery in order that there will be no need for human interaction with the physical Observatory.

When the Observatory is automated the University professors will be able to use it, and also maybe rent out the time of observing the sky, and maybe also let other Observatories use our design.

This is an abstract SDD document since many of these are still not defined, and still might be changed over time.

## 1.2 Scope

- (1) Understanding how the Observatory works.
- (2) Build the communication system of the machinery to the main manager.
- (3) Make sure data can flow in from the machinery, and that commands can be sent back to them.
- (4) Build a GUI that can show the state and actions that are done by the machinery.

## 1.3 Overview

The Observatory has specific machinery, we would like that for the start of this project we will create an abstract system that can communicate with a verity of machines and then zoom in to this Observatory specific needs.

## 1.4 Definitions and Acronyms

ASCOM- Astronomy Common Object Model

ACP- Astronomer Control Panel

SIPS- Scientific Image Processing System

DLL- Dynamic-Link Library

WPF- Windows Presentation Foundation

Dome- The housing unit to protect the telescope, it has motors to rotate and to open/close to allow the telescope to see the stars.

Mount- mechanical structure which supports a telescope

Focuser- helps the telescope focus on objects

Guider- is a camera that helps the telescope stay looking at a specific object

MVVM- Model View ViewModel

WPF- Windows Presentation Foundation

PL- Project Layer

BLL- Business Logic Layer

DAL- Data Access Layer

## 2. SYSTEM OVERVIEW

We will focus on operating the machinery within the Observatory, that includes the UI and communication system between the UI and the Observatory. We will build an Interface that can control the machinery in the Observatory and get the machinery state and data collected.

## 3. SYSTEM ARCHITECTURE

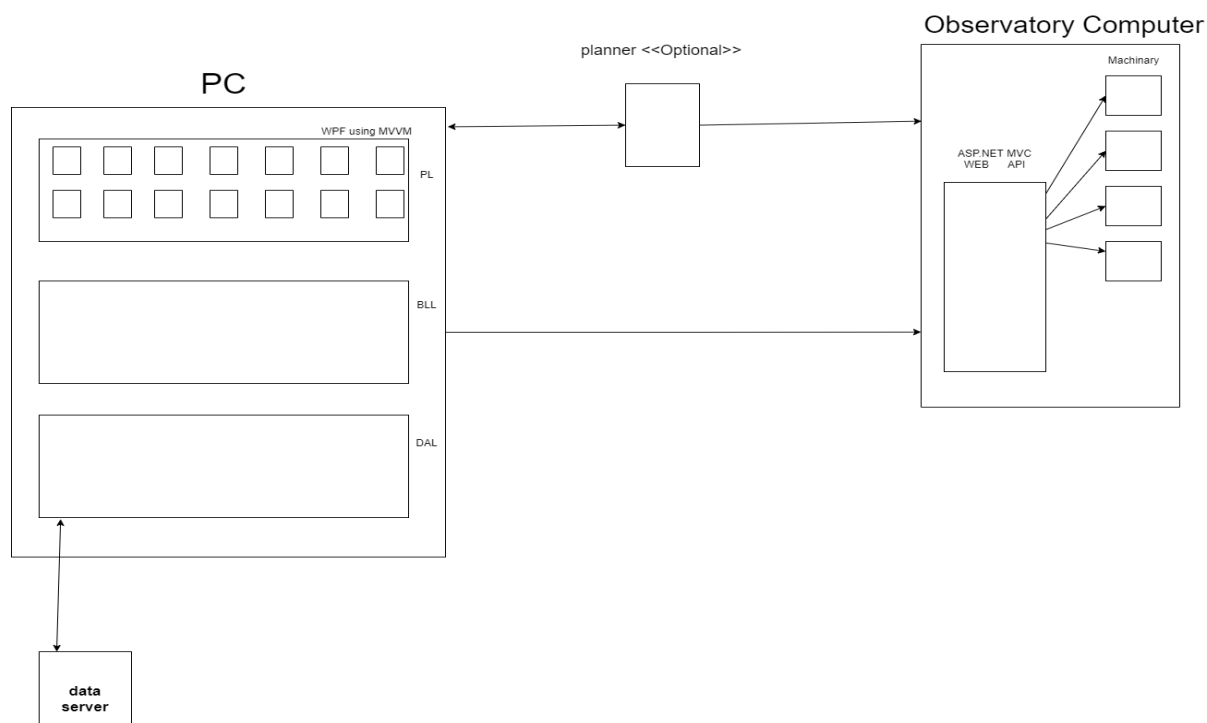
### 3.1 Architectural Design

The System will assume that the Observatory machinery is connected to the Dome Main computer, and from that computer we will communicate to the machinery. We will build a server where we will develop a WPF, from there we will send the data stored to another database storage server. And here is an attached image of the process.

As of now there is no custom way to communicate between the Observatory Computer and the machinery, so we will need to come up with a way to do that we have 2 options for now:

- 1) Build DLL files that will do the communicating with the machinery and physically install them on the Observatory Computer.
- 2) Use windows automations in order to run the commands to the machinery.

This will be decided later on after more research went into this.



### **3.2 Decomposition Description**

Observatory Main Computer – This is the computer that is sitting in the Observatory and it has its own programs, we will need to add to this computer an API that will be able to communicate with the WPF that we will build.

WPF – Our design will include 3 main pages:

PL: This Layer will be the Layer to choose the different machines in the Observatory.

BLL: This Layer will be the controls for the machine chosen showing the different options that can be included.

DAL: This Layer is in charge to communicate with both Observatory Main Computer, and the DB Storage Entity, Systems.

DB Storage – Still undecided what DB we will work with.

### **3.3 Design Rationale**

We want to have a controller from outside the Observatory so one of our big steps is to have an API communicating with the outside world. Therefore we will build this WPF on a different server than the one in the Observatory, and at the same time we will want also to have users add simple requests that then can be translated to complex movement of the Observatory, to make it as simple as possible to the users using this system.

## **4. DATA DESIGN**

### **4.1 Data Description**

We will use a Data lake structure for the Data, so that we will be able to put in all types of data without needing to normalize it. We have a high-level data and low-level data, since we are building an adaptable system. The high-level data will be an Object structure where will have the features of a data collecting option and its position and also the ability to change and get commands. While the low-level data will be the content of each Object and how it is defined and its own options.

## 4.2 Data Dictionary

Since we will use a Data lake structure it will hold the data in any way we put it in, we will need to put in the data from many sources:

- User's data – information on our clients.
- Plans – list of objects where we connect the client with their request.
- Observatory State – each machine and its state, it is optional to also store data by date(backlog).
- Images – we will need to connect the output of the Observatory which is mainly images to the client ordered and maybe also the location it took the image.

## 5. COMPONENT DESIGN

Observatory Main Computer - The Main Computer is the main brain of this whole operating system, it gathers all data from the sensors and makes decisions based on them, the Computer can also tell the machinery to move and act upon complex demands such as following a star over the night but if its rainy or cloudy close the dome, and protect yourself. The Computer will also be connected to the outside world to be able to get plans from the Users.

WPF – is made of 3 main parts:

1. PL – will be the main interface for the client, the actual design will be defined.
2. BLL – here will be the calculations and decision.
3. DAL – this will be connected to the data lake.

Db Storage – we will use the Data lake storage system, but how exactly will be defined later.

## 6. HUMAN INTERFACE DESIGN

### 6.1 Overview of User Interface

The main user will be the client to add plans, but for now the goal of the interface is to also show control of the Observatory, it will have the control for all the machinery in the Observatory, their state and their options of movement.

The Interface will be designed as an abstract view so it can be adjustable for when we add machines to be monitored by the maintenance team.

## **6.2 Screen Objects and Actions**

Let that the object displayed will perform the actions needed, we will have a manual control which is the Human override that will give the Human full action to control and move the machinery, this is in case of an unknown error.

We will also need to allow access to user input which will go to the optional Planner.