# Milestone 4 – Requirements and Preliminary Design

###### Volunteer Movie Scheduler

### Summary Use Cases

**Use Case: Producer creating the script**

**Actor: Producer**

The clicks the create script button with the intention of putting a new script into production. The producer provides the name of the script, and the system saves it into the database. The producer is then returned to the main menu and given a message indicating success.

**Use Case: Producer creating a scene:**

**Actor: Producer**

The producer looks in the script menu and clicks on the “create a scene” button with the intention of adding a new scene into production. The producer is then asked for the name of that scene, the names of the required volunteers and equipment, and a small description regarding the scene. The system then saves the scene information into the database. The producer is then sent back to the previous menu and given a message indicating whether or not the scene has been successfully added.

**Use Case: Producer removing a scene from a script:**

**Actor: Producer**

The producer wishes to remove a scene from a script and presses the remove button next to the scene he or she wishes to remove. The producer is then prompted with a message that asks if he or she is certain of the decision. If the producer presses yes, they are given a message prompting them of the successful removal of the scene from the script, the system removes the scene information from the database, and the producer is returned to the previous menu. If the producer presses cancel, he or she is returned to the previous screen and the scene is removed from the database.

**Use Case: Removing a script:**

**Actor: Producer**

To a remove a script it must be empty or all scenes within the script must be marked as completed. The producer presses the remove script button with the intention of removing the script. The system then performs a quick check is done to ensure that the above requirements are met. The producer is then presented with a menu that asks whether or not he or she is certain of the decision. If the producer presses yes, system removes the script and all of its scenes from the database. If the producer presses no, the script is not removed.

**Use Case: Completion of a scene:**

**Actor: Producer**

The producer wants to indicate that a scene has been completed. He or she must then click the checkbox next to the scene. The system then removes the scene from consideration when generating a schedule, but keeps it in the database as a reference.

**Use Case: Adding volunteer/equipment to a scene:**

**Actor: Producer**

The producer wishes to edit the requirements of a scene and presses the edit button next to the scene. The producer is then presented with a list of the scenes requirements. The producer must then click the add requirement button to add a required volunteer or piece of equipment to the scene. Once the producer has indicated the volunteer or piece of equipment, the requirement is added to the scenes requirements. The system then performs a conflict check to ensure that the newly added volunteer or piece of equipment is available when that scene is scheduled to be shot. If no conflict is detected, the producer is returned to the previous screen and given a message indicating the successful addition of the requirement. If a conflict is detected, the producer is informed that the scene can no longer be filmed at its scheduled time.

**Use Case: Removing a volunteer/equipment from a scene:**

**Actor: Producer**

The producer wishes to edit the requirements of a scene and presses the edit button next to the scene. The producer is then presented with a list of the scenes requirements. The producer must then click the remove requirement button. The producer is then asked whether or not he or she is certain of her decision. If the producer presses yes, the requirement is removed from the screen and the producer is brought back to the previous menu. If the producer selects no, then the requirement is not removed from the scene.

**Use Case: Producer creating a schedule**

**Actor: Producer**

The producer wants to begin the process of scheduling scenes. The producer opens the schedule tab and views the current schedule at a glance. The scenes that have not yet been scheduled will appear on the left in a list, and the producer will click the scene and assign it to a date, and if the scene can get scheduled there (There's no conflict) then it is.

**Use Case: Conflict Resolution**

**Actor: Producer**

The producer has a conflict, after trying to schedule a scene, so they're brought to another window which has various options for resolving the conflict. Ignore means that the scene will be scheduled anyway, and the producer will be prompted to confirm this. The other basic option will be to contact all those involved in the scene through e-mail.

**Use Case: Producer "Printing" The schedule**

**Actor: Producer**

Once all conflicts have been resolved and every scene schedule the producer wishes to create a tangible document in the form of a spreadsheet or a pdf of the schedule. On the schedule scene, the producer will have a button that says print schedule, which will only be available if the schedule is not in conflict.

**Use Case Name: Add Scene to Schedule**

**Actors: Producer**

The Producer wishes to schedule the date and time which a scene will be filmed. The Producer switches to the 'Scheduling' tab, which displays a list of all the scenes, plus a monthly calendar view centred on the current month, which displays the current shooting schedule. The Producer then Right-Clicks the appropriate date on the calendar and selects 'Schedule Scene' from the context menu which appears. This will open a small dialog window, defaulting to the date which was under focus when the Producer Right-Clicked. The Producer can then select a Scene from a list, and alter the date-time when it will be shot. The Producer can then click 'Okay' which will add the scene to that date in the schedule, or 'Cancel' to back out of the operation. After the scene is added to the schedule, the schedule is then validated to ensure that no conflicts of availability have arisen from the change in the schedule.

**Use Case Name: Remove Scene from Schedule**

**Actors: Producer**

The Producer wishes to remove a scene from its current scheduled time. The Producer switches to the 'Scheduling' tab, which displays a list of all the scenes, plus a monthly calendar view centred on the current month, which displays the current shooting schedule. The Producer then Right-Clicks on the appropriate scheduled scene that they wish to delete and selects 'Delete Scene <Scene\_Name>' from the context menu which appears. A message dialog window then appears asking the Producer to confirm deleting the scene from the schedule. If they click 'Yes' on the dialog then the Scene is removed from the schedule, otherwise no action is taken.

### Fully-Dressed Use Cases

### Supplementary Specifications

**Functionality**

**User Differentiation**

In order to protect the integrity of the system, producers and volunteers will have different privileges, which is achieved by having one main, localized, system with a separate login for the volunteer’s system.

**Modifiability**

Through the main system, the producer can make changes to the script, change the schedule, and access the contact information of all volunteers and equipment. Through the volunteer’s system, the volunteers will only be able to view the schedule and change their own contact information.

**Usability**

**Easy to Navigate**

The project should be easy to understand and navigate. This will be achieved through accurate descriptions of the actions to be performed, and an easy to use interface.

**Easy to Correct**

The project should allow for an easy correction of mistakes. It should be easy to back out of every action before it becomes permanent.

**Consistency**

The user should only be able to create and add volunteers and equipment that don’t already exist within the database (i.e. no duplicates).

**Quick generation**

The generation of the schedule should be as quick as possible and not result in a temporary freeze. The schedule should also attempt to find the schedule that allows for the filming of all of the scenes as quickly as possible.

**Easy to modify**

It should be easy to add, remove, and modify information regarding scenes, volunteers, and equipment.

**Quick conflict detection**

Conflict detection will occur every time something in the system changes; for example, a time availability of a volunteer changes. Any new conflicts should be immediately reported to the producer and suggested solutions should be readily available.

**Error information**

Any error that occurs should provide a sufficient description for the user to be able to resolve the issue.

**Supportability**

**Adaptability**

The system will be compartmentalized to allow for changes to occur without affecting the entire system.

**Future Proof**

The system should be made to allow for increased future functionality. These may include, but are not limited to, multiple scripts, and different teams of volunteers for different scripts.

**Implementation constraints**

**Language Tool**

Achievatron Unlimited is using Java technologies solution. This will facilitate portability, ease of use, support, and modifiability.

**Database Management System**

Achievatron Unlimited Will be using the object relational database management system PostgreSQL to meet the needs of the project. This will allow for a reduction in data redundancy, fast and efficient storage and retrieval of information, efficient organization of data, and a stable foundation for the organization of data

**Legal issues**

**Lost/ stolen/ damaged equipment**

Any loss of, or damage to, equipment should be removed from scheduling and documented to allow the producer to give an accurate report to the equipment owner.

**Application specific rules**

**Scene cooperation**

The volunteers and equipment selected for a scene must all share a time availability, otherwise visual feedback will be provided to indicate that this scene cannot be shot with the current time constraints.

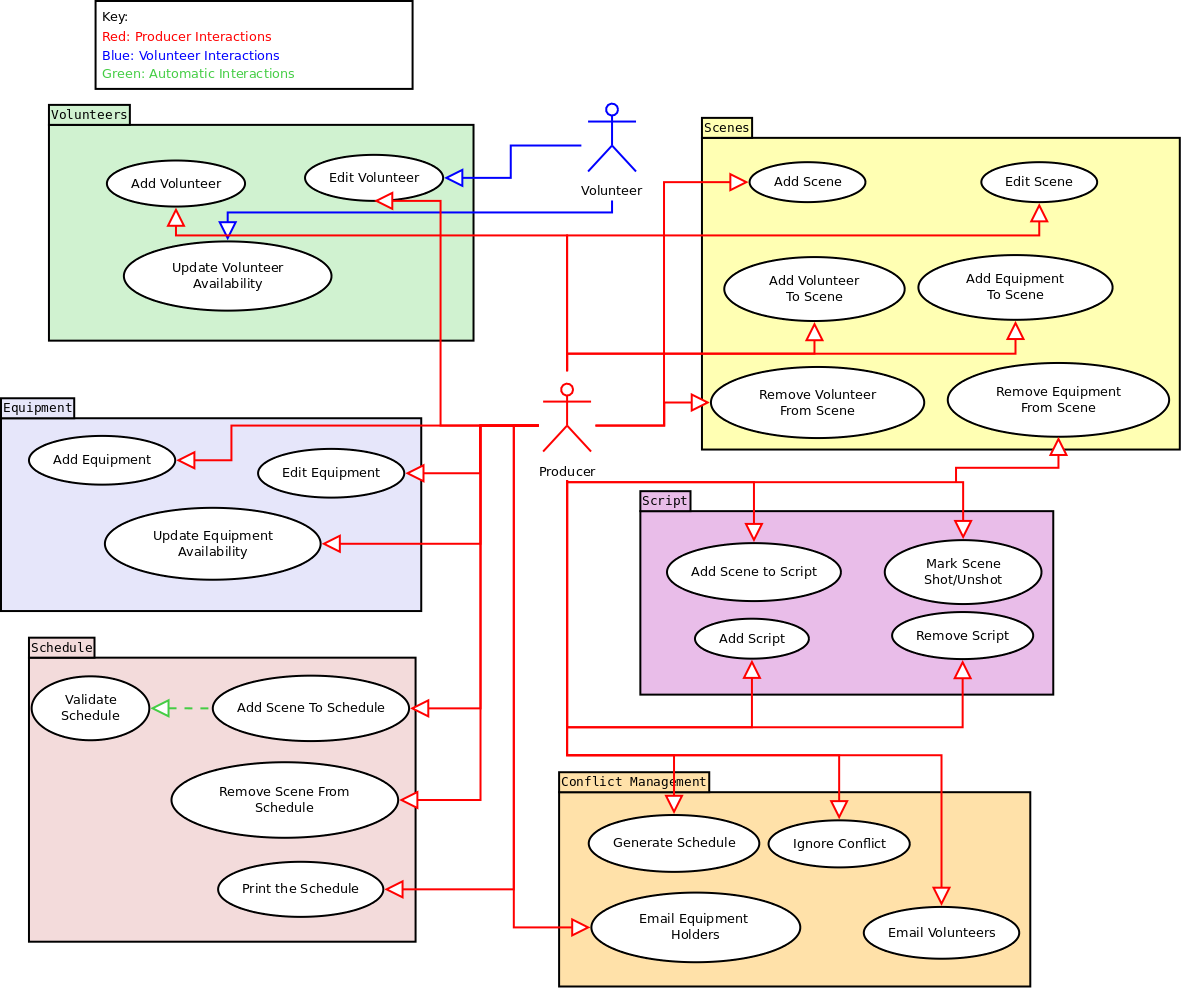
**Information in domain of interest**

**Scheduling**

We need to maintain the integrity of the schedule. If a change occurs that results in a conflict with the current schedule, the producer is notified immediately. Also, to increase security and consistency, the producer is the only person allowed to change the equipment availabilities.

The system should be able to handle different date formats as well as both military and standard time.

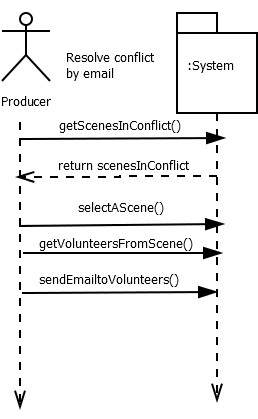
### Use Case Diagram

****

### Domain Model

### Glossary

### System sequence Diagrams



### C:\Users\Mitchell\AppData\Local\Microsoft\Windows\INetCache\Content.Word\createSchedule.png

### OPeration contracts

### project planning

Our previous project planning diagram, in an updated form:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task Description | Who? | Time Expected | Actual Time | Expected Completion | Status |
| Set up Github | Iain | 1.5h | 2h | Sept 22nd | Complete |
| Divide up work appropriately | All | .5h | .5h | Sept 22nd | Complete |
| Design the database API | Iain | 3h | 6h | Oct 1sts | Complete |
| Create example databases | Iain | 1h | .5h | Oct 1st | Complete |
| Create Buisness Objects | All | 3.5h | 5h | Oct 4th | Completed |
| UML Modelling of system | All | 2h | 2.5h | Oct 15th | Completed |
| Decide on Conventions | All | 1h | 1h | Oct 1st | Completed |
| Code: Volunteer/equipment Availability | John | 8h | 10h | Oct 31st(M4) (Primary Success) | Completed |
| Code: Scene Requirements | Ryan | 8h |  | Mid Nov | In Progress |
| Code: Schedule Generation | Iain | 10h |  | Mid Nov | In Progress |
| Code: Conflict Resolution | Mitchell | 9h |  | Mid Nov | In progress |
| Code: Applet Prototype | Matt | 7h | 10h | Oct 31st  (Primary Success) |  |
| Usability Testing | ALL |  |  |  | Not Started |
| Code: Central UI | Iain | 2h | 5h | Oct 31st | Completed |
| 4. Use cases | Ryan | 3h | 2h | Oct 31st | Completed |
| 4. Specifications |  | 2h | 1h | “ | Completed |
| 4. Use Case Diagram | Iain | 1h | 1.5h | “ | Completed |
| 4. Domain Model | John | 1.5h | 1.5h | “ | Completed |
| 4. Glossary | Matt | 1h | 2 | “ | Completed |
| 4. System Sequence Diagram | All | 4h | 6h | “ | COmpleted |
| 4. Contracts | All | 2h | 4h | “ | completed |
| 4. Implementation | All | (Derived from above) | 20h | “ | Completed |
| 4. Project Plan | Mitchell | 1h | 1h | “ | COmpleted |
| 4. Meeting Minutes | Mitchell | .25h | .25h | “ | Not Started |
| 4. Ensure proper format for hand in | Mitchell | 1h | 1h | “ | Not Started |

% Contribution

New to do:

### appendix

Meeting minutes etc