

# MILESTONE 5 UPDATED DESIGN AND NEAR-COMPLETE IMPLEMENTATION

## SYSTEM OPERATIONS

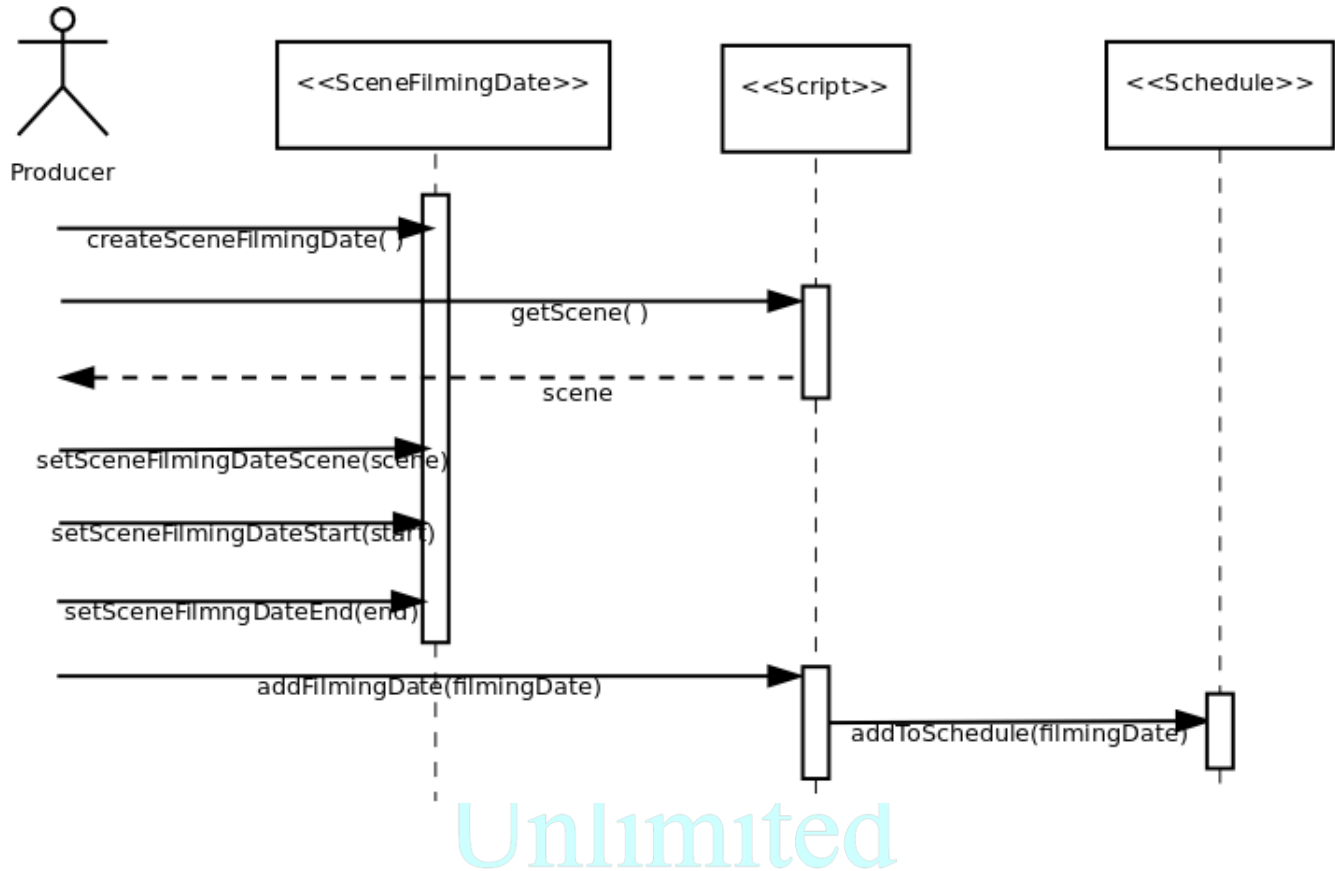
Page | 1

- setSceneName
- addVolunteer
- removeVolunteer
- addEquipment
- removeEquipment
- addSCeneToScript
- getScenesInConflict
- selectAScene
- getVolunteersFromScene
- sendEmailToVolunteers
- setUsernamePassword
- userValidated
- sendVolunteerInfo
- createNewVolunteer
- setVolunteerDetails
- addAvailability
- volunteerAdded
- getScript
- getScriptSchedule
- createSceneFilmingDate
- getSceneByName
- setfilmingDateSCene
- setTimeOfFilmingDate
- addFilmingDateToSchedule

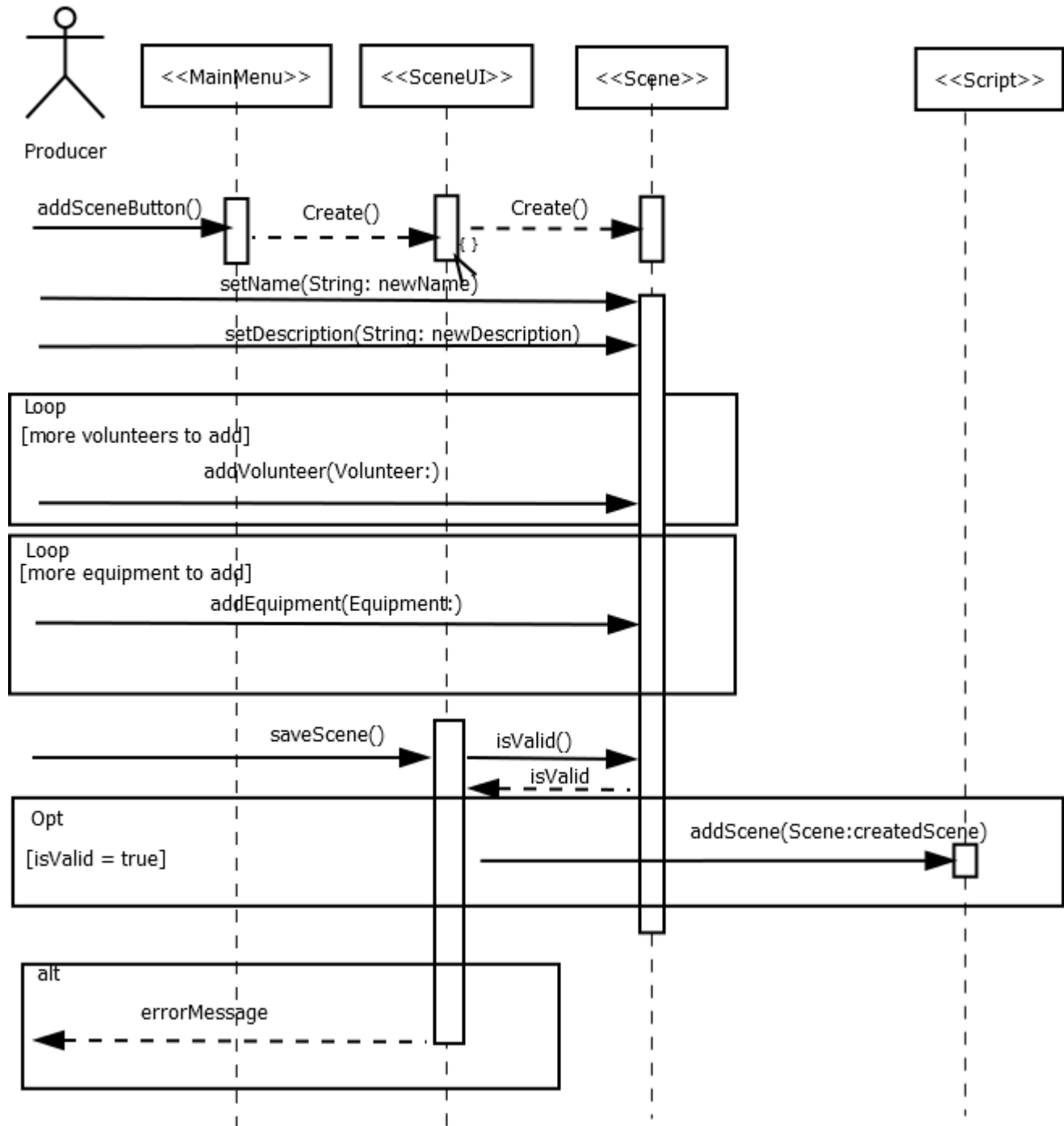


## SEQUENCE DIAGRAMS

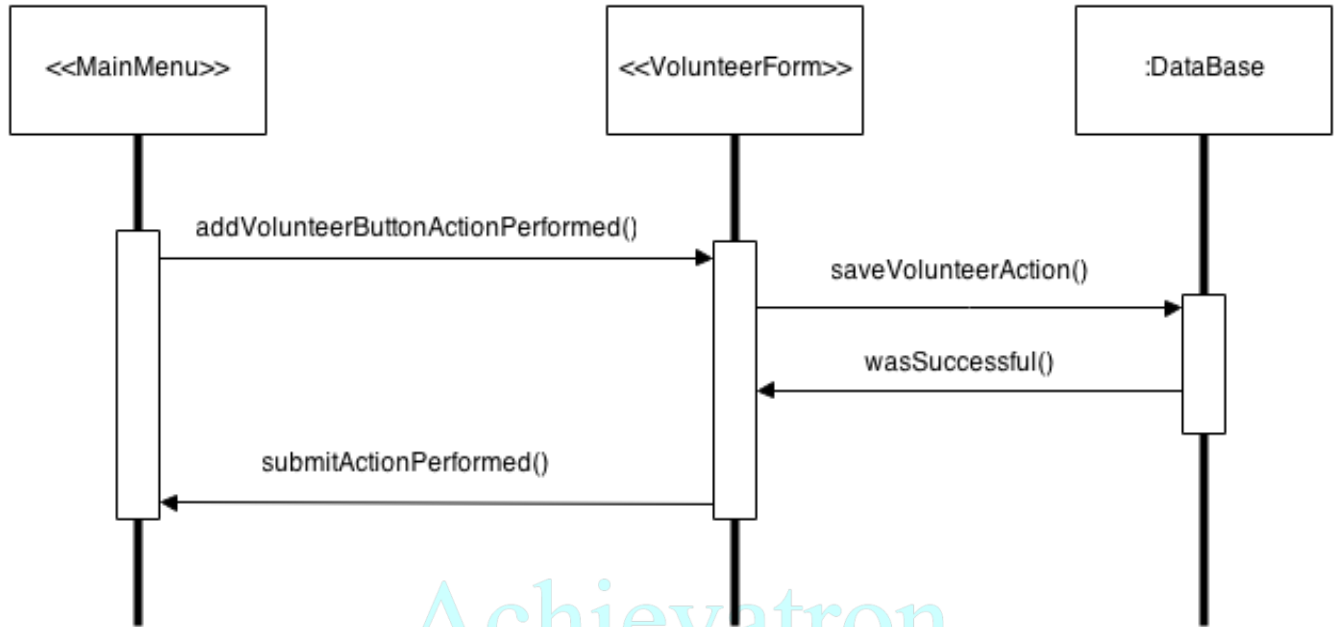
Add Filming Date To Schedule



## Add Scene



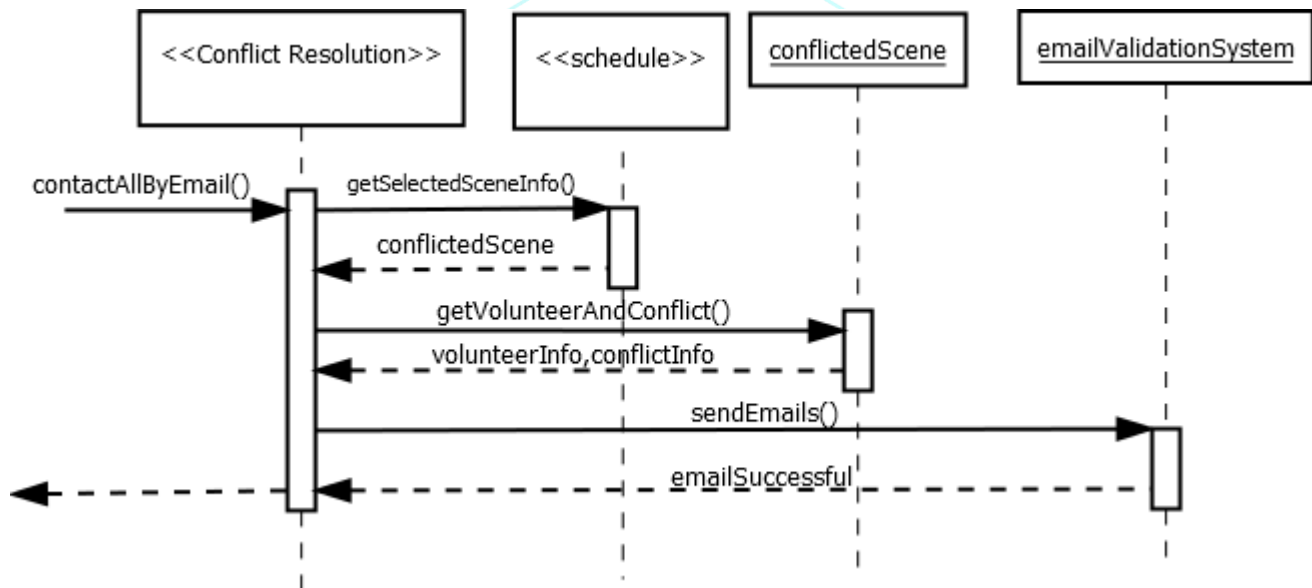
## Add Volunteer



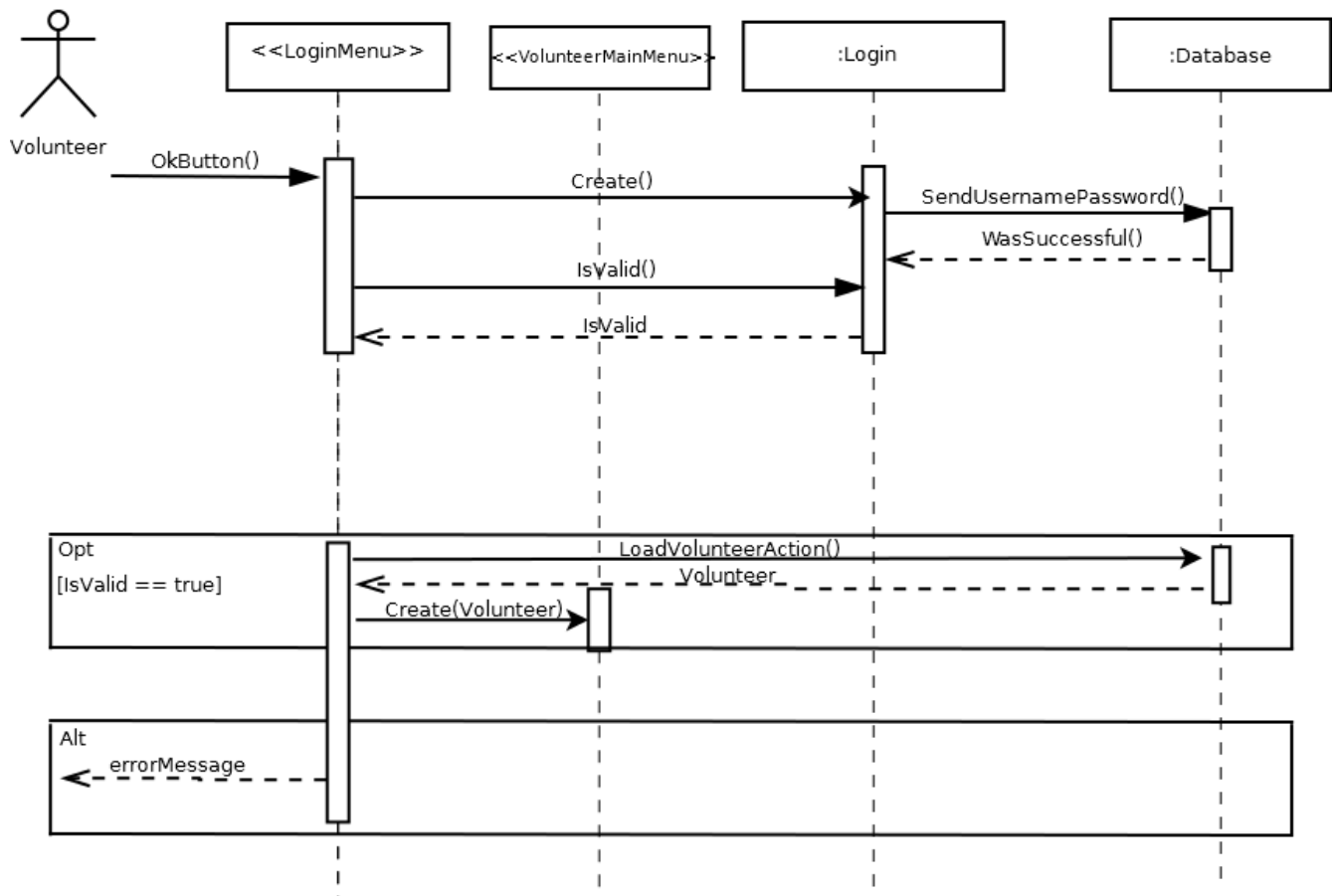
Page | 4

Achievatron

## Send Email to volunteers in conflicted Scene



## Volunteer Login



Unlimited

## CLASS DIAGRAM

See the ClassDiagram.png file in the hand in folder. It's too big to display clearly in this document.

## GIT LOG

See the file entitled "gitlog.txt" in the submission folder!

## USER MANUAL

MovieScheduler - User Manual

-----

## Table Of Contents

## 0. Introduction

1.Quick Start ~ How to run the application straight from the submission

2.Build requirements ~ What is needed to compile/run from source

3.Installing from Source ~ How to Build/Run the project from source

4.Overview ~ A Brief overview of the interface for MovieScheduler and VolunteerAvailability

5.Running the Primary Success Scenario #1 ~ Using MovieScheduler to add a Volunteer to the script/database

6.Running the Primary Success Scenario #2 ~ Using VolunteerAvailability to login as as volunteer to view Contact/Availability information

7.Running the Primary Success Scenario #3 ~ Using the MovieScheduler to add a scene to the script/database.

8.Running the Primary Success Scenario #4 ~ Using the MovieScheduler to schedule a scene.

9.Running the Primary Success Scenario #5 ~ Using the Movie Scheduler to resolve conflicts.

10.Running the Primary Success Scenario #6 ~ Using the MovieScheduler to add equipment to the script.

-----other success scenarios-----

11.Running Success Scenario #7 ~ Using the MovieScheduler to remove a scene from the script.

12.Running Success Scenario #8 ~ Using the MovieScheduler to edit a scene.

13.Running Success Scenario #9 ~ Using the MovieScheduler to remove a volunteer from the script.

14.Running Success Scenario #10 ~ Using the MovieScheduler to remove equipment from the script.

15.Running Success Scenario #11 ~ using the MovieScheduler to edit equipment in the script.

## 0. Introduction

-----

Welcome to MovieScheduler!

What's submitted:

Milestone 5 - Achievatron Unlimited.pdf ~ Our documentation for Milestone 5

MovieScheduler.zip - the built jar for the primary application, MovieScheduler

VolunteerApp.zip - the built jar for the secondary application, VolunteerApp

ClassDiagram.png – because it's too big to fit properly.

Gitlog.txt – our git log.

Primary Success Scenario 5 demonstration.swf and .htm – A flash video demonstrating the success scenario 5, since it's a bit difficult to do

\*note\* - This project has two applications, as listed above. They are built from the same source, but a different main class path.

MovieScheduler uses moviescheduler.MovieScheduler.java as the main class, while VolunteerApp uses volunteeravailability.VolunteerAvailability.java as the main class.



## 1. Quick Start

-----

The latest builds for both MovieScheduler and VolunteerAvailability are included in the assignment submission. MovieScheduler is in MovieScheduler.zip, and VolunteerAvailability is in VolunteerApp.zip.

To run either application, simply extract the respective zip file, and double click the respective application's .jar file and the applications will run.

## 2. Build requirements

-----

\* Java SE 1.8, JDK 8 ( <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> )

\* PostgreSQL JDBC driver ( <http://jdbc.postgresql.org/> )

\* JUnit 4.10 (<http://junit.org/> )

### 3. Installing from Source

Included in our project submission is the source code for the project. Building is possible through Netbeans >=8.0.

If building / running from the source, please make sure the Build requirements above are met. Following these instructions should result in a seamless build of both project applications. However, to make your time easier, we have both of the latest builds submitted.

The steps for installation from the source (Using Netbeans) are the same for Windows, Mac, and Unix.

#### 0. Unzip src.zip included in our submission

##### 1. Open Netbeans

##### 2. Select File -> Open Project

##### 3. Browse for the MovieScheduler netbeans project from src.zip and select it, opening it as a project in Netbeans.

##### 4. To build / run the primary MovieScheduler application:

###### 4.1. Building a .jar from Netbeans:

###### 4.1.1. Right click Movie Scheduler and select Properties

###### 4.1.2. Under Build-> Packaging make sure 'Compress JAR File' is selected, then click 'OK'.

4.1.3. Select Run->Clean and Build Project. MovieScheduler.jar will be created in the project folder/dist/\* along with required libraries.

###### 4.2. Running from Netbeans:

###### 4.2.1. Right click Movie Scheduler and select Run.

###### 4.2.2. The application should launch straight from netbeans.

##### 5. To build / run the secondary VolunteerAvailability application:

###### 5.1. Building a .jar from Netbeans:



5.1.1.Right click Movie Scheduler and select Properties

5.1.2.Under Run, change the 'Main Class:' to volunteeravailability.VolunteerAvailability.

5.1.2.Under Build-> Packaging make sure 'Compress JAR File' is selected.

5.1.3.Click 'OK'.

Page | 9

5.1.4.Select Run->Clean and Build Project. MovieScheduler.jar will be created in the project folder/dist/\* along with requiried libraries. This is the secondary

VolunteerAvailability application.

\*note\* Because both applications are built from the same main project, MovieScheduler, the jar created for both is called MovieScheduler.jar.

our submission the VolunteerAvaialbility main class build is in VolunteerApp.zip, renamed appropriately.

5.2.Running from Netbeans:

5.2.1. Expand MovieScheduler/Source Packages.

5.2.2. Expand volunteeravailability.

5.2.3. Right click VolunteerAvailability.java and select Run.

5.2.4. The application should launch straight from Netbeans.

#### 4. Overview

-----

The following is a introduction to the interface of both applications:

##### 4.1. MovieScheduler

-----

When you run MovieScheduler, you are first shown a welcome screen. It asks for the name of a script, which you can enter, and then click 'Okay'. If the script is blank, an appropriate error display is shown. This transitions into the MainMenu with the script name you have provided.

There are two tabs - Script and Schedule.

The Script tab - this is the area where you can add volunteers, equipment, and scenes to the script.

Volunteer section -

- The top part of this section shows the Volunteers that are in the script, so, when the script is first initialized the box will display "No volunteers currently in script".

- The bottom part of this section has options to Add, Edit, and Remove a volunteer.

- the Add Volunteer button brings up a Volunteer Signup sheet form where you enter the volunteer's contact info and availability and submit it, adding the information to the database / the script that is being worked on.

- the Edit Volunteer button when implemented will allow the volunteer that is selected to be edited, i.e. changing contact / availability information

- The Remove Volunteer button when implemented will allow a volunteer that is selected to be removed from the script/database

Equipment - This middle section of the Script tab shows information about the Equipment in the script, and lets you select an Equipment to view addition information. There are also buttons that when implemented will allow adding, editing, and removing equipment from the script.

Scene - The top part of this section shows the Scenes that are in the script, and allows you to check off whether the scene is scheduled / completed. The "View Requirements" button when implemented will allow the producer to view the requirements of the scene selected (e.g. cast members needed, equipment needed, etc).

The Schedule tab - this will be an area where you can produce the schedule and resolve conflicts in it.

#### 4.2. VolunteerAvailability

-----

When you run the VolunteerApp, you are first shown a Login screen. It has a Username and Password field, where the Username will be the e-mail of the volunteer.

There is also a Forgot Password? section that will be implemented, creating a form with an email field to reset and email the password of the volunteer.

After logging in, there is currently a display showing the Contact Information, and Availability. At the bottom, there is a logout button, which terminates the application.

## 5. Running the Primary Success Scenario #1

-----

Using MovieScheduler to add a Volunteer to the script/database.

This primary success scenario occurs on the primary application, MovieScheduler (The pre-made build in MovieScheduler.zip, or made from source (3.- instructions #4))

In this success scenario, the Producer creates a script, and is able to add volunteers to it.

Instructions:

1. The producer opens MovieScheduler application
2. The producer chooses a script name, and clicks 'Okay'. If the script name is blank, an appropriate error dialog is shown.

For each volunteer the producer adds to the script, the following happens:

3. At the bottom left of the Script tab under Volunteer, the producer clicks "Add Volunteer". A new window is created with a Volunteer Signup Sheet form.
4. The producer adds the contact information in the text fields corresponding to the first name, surname, phone, e-mail.
5. The producer enters availability start and end times, and clicks "Add" to add them to the volunteer.
6. The producer is done adding the information, and clicks "Submit".

Finally, the producer can view the volunteers they added to the script:.

7. To view the volunteers the producer clicks the dropdown box below "View Availabilities" in the volunteer section, and selects the volunteer of interest.

## 6. Running the Primary Success Scenario #2:

-----

Using VolunteerAvailability to login as as volunteer to view Contact/Availability information

This primary success scenario occurs on the secondary application, VolunteerAvailability (The pre-made build in VolunteerApp.zip, or made from source (3. - instructions #5))

In this success scenario, a Volunteer successfully logs in to the VolunteerAvailability application to view their contact / availability info on file.

1. The volunteer opens the VolunteerApp application
2. On the Login Screen, the volunteer enters their e-mail and password credentials. If either field is left blank, an appropriate error message is displayed.

USERNAME: Passme [case sensitive]

PASSWORD: password

3. If the login was not successful, an appropriate error message is displayed notifying the volunteer, and they can try again.
4. If the login was successful, the main screen of the VolunteerAvailability application is initialized with that volunteers Information.
5. The volunteer sees the information they have on file, and clicks "Logout", ending the session.

## 7. Running the Primary Success Scenario #3

-----

Using the MovieScheduler to add a scene to the script/database

This primary success scenario occurs on the primary application, MovieScheduler.

In this success scenario, the producer clicks on the add button beside the scene list on the main menu, inputs all of the necessary information through the scene UI,

and saves the scene to the script and to the database.

1. The producer opens up the main menu
2. The producer clicks the add button beside the list of scenes in the first tab.
3. The producer is presented with the SceneUI screen.
4. The producer is presented boxes to insert the scene information to, as well as a list of all of the volunteers and equipment available in the script.
5. The producer enters information about the scene including the name, description, and volunteers and equipment.
6. The user presses the submit button.
7. The scene is saved to the database and to the script.

#### 8. Running primary success scenario #4 Setting a scene filming date

-----  
Using the MovieScheduler to schedule a scene production time.

This primary success scenario occurs on the primary application, MovieScheduler.

In this success scenario, the producer opens up the application to view the main menu, clicks on the secondary tab(schedule) where he or she is presented with a calendar.

The producer must then right click one of the scenes in the list on the left and click the schedule button in the context menu. The producer is then presented with a screen

that allows him or to choose period in which to schedule that scene. After the producer clicks ok, the scene is scheduled for that date, which is reflected in the calendar.

1. The producer opens up the main menu.
2. The producer clicks the on the secondary, schedule tab.
3. The producer right clicks the scene that he or she wishes to schedule.
4. The producer clicks schedule in the context menu.

5. The producer inputs the time period in which to film this scene.
6. The producer clicks ok and the scene is scheduled for that date.

9. Running primary success scenario #5 Viewing and resolving a conflict.

-----

Using the MovieScheduler to schedule and resolve a conflict.

A conflict is a state in which a scene has been scheduled at a time when it cannot be filmed.(Invalid filming date).

In this success scenario, The producer opens up the application to view the main menu, switches to the schedule tab, clicks on the conflict button, and clicks on a scene to view its conflicts.

If the scene has a conflict, the producer can email all of those involved with the scene or edit the scene to resolve the conflict.

NOTE: this success scenario is difficult to perform because it requires real email addresses, and other aspects that are beyond test data to perform.

As a result, our member Mitchell has included a video labelled, primary success scenario # 5, which walks you through the scenario to show how it works.

1. The producer opens up the main menu.
2. The producer clicks on the secondary, schedule tab.
3. The producer clicks the conflicts button and is presented with the conflict screen.
4. The producer can then click on a scene to see all of the conflicts that are involved with it.
5. If the selected scene has any conflicts, the producer is presented with a list of those conflicts.
6. The producer may then email the members involved, or edit the scene to resolve the conflict.

10. Running the Primary Success Scenario #6 ~ Using the MovieScheduler to add equipment to the script.

-----

Using the MovieScheduler to add a piece of equipment to the script and database.

- 1.The producer opens up the main menu.
- 2.The producer clicks on the add button next to the equipment list.
- 3.The add equipment ui opens up.
- 4.The producer enters all of the informmation regarding the equipment as well as the time periods in which the equipment is available.
- 5.The producer presses the submit button.
- 6.The equipment is saved to the script and database.

-----other success scenarios-----

- 11.Running Success Scenario #7 ~ Using the MovieScheduler to remove a scene from the script.

-----

Using the MovieScheduler to remove a scene from the script and database.

- 1.The producer opens up the main menu.
- 2.The producer presses the remove button next to the scene list.
- 3.The scene is removed from the script and from the database.

- 12.Running Success Scenario #8 ~ Using the MovieScheduler to edit a scene.

-----

Using the MovieScheduler to edit a scene.

- 1.The producer opens up the main menu.
- 2.The producer clicks the edit button next to the scene list.
- 3.The Scene menu will open up with all of the selected scenes information already listed.
- 4.The producer edits the scene name, description, volunteers, equipment or a combination of the two.
- 5.The producer presses the submit button.

6.The scene is saved to the database and the script.

13.Running Success Scenario #9 ~ Using the MovieScheduler to remove a volunteer from the script.

Page | 16

-----  
Using the MovieScheduler to remove a volunteer.

- 1.The producer opens up the main menu.
- 2.The producer selects a volunteer.
- 3.The producer clicks on the remove button beside the volunteer list.
- 4.The volunteer is removed from the script and database.

14.Running Success Scenario #10 ~ Using the MovieScheduler to remove equipment from the script.

-----  
Using the MovieScheduler to remove equipment.

- 1.The producer opens up the main menu.
- 2.The producer selects the equipment he or she wants to remove.
- 3.The producer clicks the remove button beside the equipment list.
- 4.The equipment is removed from the script and database.

15.Running Success Scenario #11 ~ using the MovieScheduler to edit equipment in the script.

-----  
Using the MovieScheduler to edit equipment.

- 1.The producer opens up the main menu.



- 2.The producer selects a piece of equipment from the equipment list.
- 3.The producer clicks the edit button next to the equipment list.
- 4.The producer edits the values of the equipment.
- 5.The producer clicks submit and the new equipment values are written to the script and database.

## MEETING MINUTES

### meeting minutes Nov 4th

discussed the possibility of using html within  
swing\java

present: Mitchell, Iain, Ryan, John

time: 5:00-5:30 present: Mitchell, Iain, Ryan, John,  
Matt

time: 4:00-5:00

discussed midterm

discussed milestone 5

went over previous test

all issues to be completed have been  
uploaded to issues on github

### meeting minutes Nov 18th

agreed on work allocation issues section on  
github

present: Mitchell, Iain, Ryan, John, Matt

agreed on coding format (get and set)

time: 4:00-5:00

agreed to implement cloneable (deep clone)  
and overwrite the original with the clone on a  
successful change

discussed milestone 5

discussed possible database fails during the  
middle of a group of transactions

made class diagram

start with begin and if anything goes  
wrong you do a rollback and try again if success  
commit

agreed on a pretty printing style

discussed grasp patterns

solidified success scenarios

discussed volunteer availabilities

discussed new main menu layout

## PROJECT PLAN

Milestone 5 Requirements:

Requirement	Who?	Expected time	Real time	Completion Date	Completed?
Sequence Diagrams	All	10h	2h (ryan)	Oct 14th	Completed
Class Diagram	Iain	3h		Nov 21 <sup>st</sup>	Completed
Implementation	All	~80h	30hr(ryan)	Nov21st	Completed
Project Plan	Mitchell				Completed

Implementation expectations:

Implementation Goal	Who's completed it for their section/Responsible	Expected time	Actual Time	Completion date	Completed?
Major Success Scenario	Matt/John/Ryan	60-70h total	80hr	Nov 21st	Completed
UI	Iain, Mitchell, Ryan, John	20h total	21h	"	Completed
Naming Conventions	Iain,Ryan	2h	3 1/2hr	"	Completed
Comments	All	5h	4h	"	"
Consistent Style	Iain	4h	2h	"	"
User Manual	Matt/Ryan	2h	2h	"	"
Git Log	Iain	A few minutes	A FEW SECONDS	"	"

## % CONTRIBUTION

Iain – 20%

Mitchell – 20%

Matt – 20%

Ryan – 20%

John –20%

## WORK REQUIRED TO COMPLETE THE PROJECT

M6:

Page | 19

What?	Who?	Expected Time	Actual Time	Required date	Completed?
Abstract	All	1h			
Intro(M2)	Mitchell	2h			
Requirements/Early Design(M3)		2h			
Updated design and Testing	All	20h			
Reengineering	All	5h			
Complete implementation (m5ish)	All	20h			
Project plan etc	Mitchell	2h			
Conclusion	Mitchel	1h			

