

# Web Application for the Junior Film Fleadh

---

## Software Requirements Specification

Version 1.0

18 Oct 2016

James Quaife  
Andrew East  
Chika Onyia  
David O'Dea

Prepared for CT216 - Software Engineering I

## Table of Contents

1.	Introduction .....	5
1.1.	Purpose.....	5
1.2.	Scope.....	5
1.3.	References.....	6
1.4.	Overview .....	6
1.4.1.	Document Organisation .....	6
2.	Specific Requirements .....	6
2.1.	Interface Requirements – User Interfaces .....	7
2.1.1.	Create Account .....	7
2.1.2.	Film Submission .....	8
2.1.3.	Staff Page – Film Review Queue.....	9
2.1.4.	Staff Page – Review a Film .....	10
2.1.5.	Admin Page – Manage Users.....	11
2.1.6.	Admin Page – Manage Submission .....	12
2.1.7.	Admin Page – Create Schedule.....	13
2.1.8.	Admin Page – Manage Bookings .....	14
2.1.9.	Film Booking - Overview .....	15
2.1.10.	Film Booking - Details .....	16
2.2.	Functional Requirements .....	17
2.2.1.	Account Creation .....	17
2.2.2.	Users Management .....	17
2.2.3.	Film Submission .....	18
2.2.4.	Submission Review.....	19
2.2.5.	Submissions Management .....	19
2.2.6.	Festival Schedule Creation .....	20
2.2.7.	Bookings Management .....	20
2.2.8.	Film Bookings and Payment.....	21
2.3.	Use Cases & Use Case Diagrams.....	22
2.3.1.	Use Case Diagrams .....	22
2.3.2.	Use Case Description: Create Login.....	23
2.3.3.	Use Case Description: Submit Film.....	23
2.3.4.	Use Case Description: Review one Film .....	24

2.3.5.	Use Case Description: Approve or Reject Films .....	24
2.3.6.	Use Case Description: Book a Showing .....	25
2.4.	Activity Diagrams .....	27
2.4.1.	Activity Diagram for Use Case #1 .....	27
2.4.2.	Activity Diagram for Use Case #2.....	28
2.4.3.	Activity Diagram for Use Case #3.....	29
2.4.4.	Activity Diagram for Use Case #4.....	30
2.4.5.	Activity Diagram for Use Case #5.....	31
2.5.	Data Flow Diagrams .....	33
2.5.1.	Context Diagram.....	33
2.5.2.	Data Flow Diagram: Level 0 Diagram.....	34
2.5.3.	Data Flow Diagram: Figure 1 .....	35
2.5.4.	Data Flow Diagram: Figure 2 .....	36
2.5.5.	Data Flow Diagram: Figure 3 .....	36
2.6.	Database Schema (JSON) .....	37
2.6.1.	Film Submission Collection.....	37
2.6.2.	Screening Calendar Collection .....	37
2.6.3.	Users Collection.....	37
2.6.4.	Bookings Collection .....	38
3.	Project Timeline.....	39
A.	Appendices .....	42
A.1	Appendix 1 – Link to GitHub Repository .....	42
A.2.	Appendix 2 – Short Requirements Document.....	42

*This page has been left blank intentionally.*

# 1. Introduction

The aim of this document is to provide analysis and insight into the complete software for the Web Application for the Junior Film Fleadh. It details the needs of our client as well as a high-level view of product features.

## 1.1. Purpose

The purpose of this document is to inspect all ideas that define the system and its requirements to the client. We shall anticipate and outline concepts that may be considered for development later but may be discarded as we progress. The document will describe our system's processes, user interfaces and layout, data flows and schema, and product development timeline. It defines the minimal viable product that we will need to create to deliver to our client. It will also define our vision of how we, the client, and the end user will view the various functionality of the finished product.

## 1.2. Scope

Primarily, the scope of this project will cater to the needs of the Junior Film Fleadh staff and management. The client's need focuses mainly on the end users' interaction with the product, and the central media content management system for the client. Our software will allow filmmakers to submit films, staff to review those films, and attendees to book tickets to view films at the festival. This is the minimum viable product for this application to be a success.

To accomplish this scope, the Meteor JavaScript framework will be the backbone of this application as it will provide real-time updates for our web client. We will use MongoDB to provide the data storage, allowing users will be able to create profiles using either locally stored sign-on credentials or a credential token from an OAuth-providing 3<sup>rd</sup> party web site, submit movie, and browse movie screenings to buy tickets. The application will provide a streamlined service to our client and the users, allowing large amounts of film submissions.

### Objectives and goals defining scope of product:

- The application will provide:
  - A single unified and streamlined platform for submissions
  - An online platform for Junior Film Fleadh staff to review submissions and submit comments
  - A tool to schedule films during the festival weeks in the various venues
  - An online store to view the films showing and purchase tickets
  - The ability to manage ticket sold during the festival
- The application will not:
  - Be a complete content management system or blogging platform
  - Provide a discussion space for end users.

### 1.3. References

**Staff of the Junior Film Fleadh:** The requirements for this application were provided by a staff member of the film Fleadh. Their information was referenced to create the scope of our project.

**Short Description Document:** Much of the planning of the scope of this project was done in creation of our initial short description document, included as Appendix 2. We have revised our scope slightly since then, focusing on a minimum viable product of filmmakers submitting films, staff reviewing films, and attendees booking tickets to films. As such, this document contained within the appendix should not be considered a part of the overall set of software requirements specification.

**Link to GitHub repository:** We will be working with git as a distributed software versioning system, and all code will be checked into GitHub. Each team member has established separate repositories forked from a single upstream. The upstream repo is being managed by one member of our team, and all completed feature branches will originate pull requests before being integrated it into the main repo. This will allow us to conduct code reviews and make comments on the feature. The link is provided as Appendix 1 (the repository is not a part of our software requirements specification).

### 1.4. Overview

The remainder of this document details the processes that we have identified will be necessary to implement the application. It illustrates the client needs that were identified in our initial and subsequent meetings with the client. It also details the current challenges facing the JFF in regards to potential growth opportunities.

#### 1.4.1. Document Organisation

The following pages in Section 2 provide an overview of the specific requirements of the project. Section 2.1 will detail the user interface requirements such as layouts, routes, and forms. Section 2.2 will detail the different functional requirements and their data inputs, processing, and outputs. Section 2.3 will detail the different use cases for each interaction the various users of the application, illustrated by use case diagrams and use case descriptions. Section 2.4 will provide activity diagrams corresponding to the use cases that we foresee occurring on the platform. Section 2.5 will illustrate our proposed database schema in MongoDB utilising JSON concepts. Section 3 will give our project timeline, and our appendix provides a referenced document.

## 2. Specific Requirements

This section details the requirements for the web application. It will be organised by the features which were decided upon during our meetings with the client and among our developers.

## 2.1. Interface Requirements – User Interfaces

### 2.1.1. Create Account

The screenshot shows a web browser window titled "JFF /signup". The page features a red header with the text "Junior Film Fleadh" and buttons for "Log In" and "Sign Up". Below the header is a black banner with the text "JUNIOR FILM FLEADH 9TH - 12TH NOVEMBER 2016". A navigation bar contains links for "Home", "Blog", "Tickets", and "Button 4". The main heading is "Sign Up". The form includes input fields for "Name", "Email", and "Contact No.". There are three checkboxes for user roles: "Student" (checked), "Teacher", and "Astronaut". A fourth checkbox, "I have read JFF's terms and privacy policy", is also checked. At the bottom of the form are "Cancel" and "Submit" buttons. A footer navigation bar contains links for "About", "Contact", "Privacy", and "Terms".

JFF /signup

Junior Film Fleadh Log In Sign Up

JUNIOR FILM FLEADH 9TH - 12TH NOVEMBER 2016

Home Blog Tickets Button 4

# Sign Up

Name

Email

Contact No.

☒ Student

☐ Teacher

☐ Astronaut

☒ I have read JFF's terms and privacy policy

Cancel Submit

About Contact Privacy Terms

## 2.1.2. Film Submission

JFF /home

Junior Film Fleadh

UserName

JUNIOR FILM FLEADH 9TH - 12TH NOVEMBER 2016

Reviews

Bookings

Title

Genre

Film Length

▼

Short

Fetaure

Short filmDescription (< 250 Chars)

Additional Comments (< 250 Chars)

☒ I have read JFF's terms and privacy policy

Cancel

Submit

About

Contact

Privacy

Terms



### 2.1.3. Staff Page – Film Review Queue

JFF /home

**Junior Film Fleadh**

JUNIOR FILM FLEADH 9TH - 12TH NOVEMBER 2016

Reviews

Bookings

Thumbnail	Title	Submitter Details	Length	Genre	Description	Comments	Reviewed status

## 2.1.4. Staff Page – Review a Film

JFF /home

Junior Film Fleadh
 

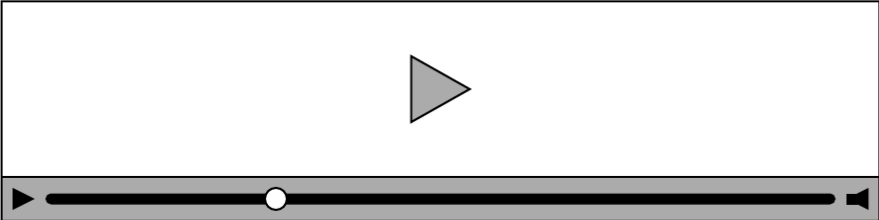
UserName

JUNIOR FILM FLEADH 9TH - 12TH NOVEMBER 2016

Reviews

Bookings

Calendar



Comments

Rate Film

Like

Neutral

Dislike

If Like or Neutral -  
Enable Cal & Venues

Suggest Time/Date

◀ March 2010 ▶

S	M	T	W	Th	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Venues	Times

Cancel

Update

## 2.1.5. Admin Page – Manage Users

JFF /home

Junior Film Fleadh

UserName

JUNIOR FILM FLEADH 9TH - 12TH NOVEMBER 2016

Home

Bookings

Schedule

Users

Collections

submissions

bookngs

users

collectionD

collectionE

collection

Users	Email	Seen	Priveleges
			User <input type="button" value="▼"/>
			Reviewer
			Admin

## 2.1.6. Admin Page – Manage Submission

JFF /home \_ □ ×

Junior Film Fleadh

UserName

JUNIOR FILM FLEADH 9TH - 12TH NOVEMBER 2016

Home

Bookings

Schedule

Users

Collections	Thumbnail	Title	Submitter	Length	Genre	Description	Comments	Reviewed status	Add to Queue
submissions									
bookngs									<input checked="" type="checkbox"/> <input type="button" value="▼"/>
users									<input type="button" value="🗑"/>
collectionD									
collectionE									
collection									

## 2.1.7. Admin Page – Create Schedule

JFF /home

Junior Film Fleadh

UserName

JUNIOR FILM FLEADH 9TH - 12TH NOVEMBER 2016

Home

Bookings

Schedule

Users

Collections

submissions

bookngs

users

Schedule

collectionE

collection

.jpg

.jpg

.jpg

.jpg

.jpg

.jpg

.jpg

.jpg

.jpg

March 2010

S	M	T	W	Th	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Reset

Confirm

## 2.1.8. Admin Page – Manage Bookings

JFF /home

Junior Film Fleadh
UserName

JUNIOR FILM FLEADH 9TH - 12TH NOVEMBER 2016

Home

Bookings

Schedule

Users

Collections

submissions










bookngs

users

collectionD

collectionE

collection

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
	 Upload Image			 Upload Image			 Upload Image
	Movie Details			Movie Details			Movie Details
	Times/Venues			Times/Venues			Times/Venues
	Prices			Prices			Prices
	 Upload Image			 Upload Image			 Upload Image
	Movie Details			Movie Details			Movie Details
	Times/Venues			Times/Venues			Times/Venues
	Prices			Prices			Prices
	 Upload Image			 Upload Image			 Upload Image
	Movie Details			Movie Details			Movie Details
	Times/Venues			Times/Venues			Times/Venues
	Prices			Prices			Prices

## 2.1.9. Film Booking - Overview

JFF /home

Junior Film Fleadh

UserName

JUNIOR FILM FLEADH 9TH - 12TH NOVEMBER 2016

Reviews

Bookings

Calendar

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

Sunday

Movie Details

Times/Venues

Movie Details

Times/Venues

Movie Details

Times/Venues

Movie Details

Times/Venues

Movie Details

Times/Venues

Movie Details

Times/Venues

Movie Details

Times/Venues

Movie Details

Times/Venues

Movie Details

Times/Venues

About

Contact

Privacy

Terms

### 2.1.10. Film Booking - Details

JFF /home

Junior Film Fleadh

UserName

JUNIOR FILM FLEADH 9TH - 12TH NOVEMBER 2016

Reviews

Bookings

Calendar

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

Sunday

Details

Type	QTY	COST	SUBTOTAL	
			Total €	

Cancel

Book

About

Contact

Privacy

Terms



## 2.2. Functional Requirements

### 2.2.1. Account Creation

#### 2.2.1.1. Introduction

The application will allow a new visitor to the web site to create a user account. They must do this before they can submit a film.

#### 2.2.1.2. Inputs

- The system will take the user's name.
- The system will take the user's email.
- The system will take the user's contact number.
- The system will take the user's account type, e.g. student, teacher, astronaut.
- The system will confirm whether user has accepted or denied the terms of service.

#### 2.2.1.3. Processing

- The above variables are taken in by the system and stored in the Users collection in MongoDB for future access.

#### 2.2.1.4. Outputs

- The system generates an email for confirmation to be sent to the specified user's address confirming the above details.

#### 2.2.1.5. Error Handling

- Ensuring the email entered follows an acceptable format for the system.
- Ensures that to proceed with account creation the terms of service are acknowledged.

### 2.2.2. Users Management

#### 2.2.2.1. Introduction

The application will allow the administrator to view all user accounts that have been created, manage their details, and assign privilege levels to staff or administrators.

#### 2.2.2.2. Inputs

- The system will draw the inputs from the Users collection, based on the inputs from 2.2.1: Account Creation.

#### 2.2.2.3. Processing

- System will allow admin to set different privileges (e.g. staff reviewer, administrator, or user) for each user account.
- System will allow admin to alter details, e.g. email address, for each user.

#### 2.2.2.4. *Outputs*

- System will capture any of the above alterations and reflect them in the Users collection.

#### 2.2.2.5. *Error handling*

- The system will confirm with the admin when assigning special privileges to a user, e.g. a confirmation dialog verifying that a selected user may be granted admin privileges.
- The system will ensure that if changing details, e.g. email address, that these details follow the correct format.

### 2.2.3. **Film Submission**

#### 2.2.3.1. *Introduction*

The application will allow a user to submit a film for entry into the festival. They will record the details about their film, and select a media file to upload.

#### 2.2.3.2. *Inputs*

- The system will take a media file from the user's computer.
- The system will take the submission's title.
- The system will take the submission's genre.
- The system will take the submission's length, i.e. short or feature length film.
- The system will take the submission's short description.
- The system will take the submission's additional comments from the filmmaker.
- The system will take confirm if the user has accepted or denied the terms of service and privacy policy for film submission.
- The system will use details from the Users collection, based on the inputs from 2.2.1: Account Creation.

#### 2.2.3.3. *Processing*

- The media file will be uploaded to the server, and possibly converted to a standardised format.
- The film metadata is saved to the Film Submissions collection, along with a link to the media file on the server's storage.

#### 2.2.3.4. *Outputs*

- The above is outputted to the Film Submissions collection in MongoDB.
- The user will receive email confirmation of the submission.
- The system will notify an admin of the new submission.
- The system will link the submission to the submitters profile stored in the Users Collection.

#### 2.2.3.5. *Error handling*

- The system will ensure the uploaded file is of an acceptable format and size.

## 2.2.4. Submission Review

### 2.2.4.1. Introduction

The application will allow a staff member to view a film streamed from the server, write comments about it, give it a rating, and propose a scheduled time slot.

### 2.2.4.2. Inputs

- The system will generate a stream for the playback of the selected film from the Film Submission collection's metadata.
- The system will take in comments from the staff reviewer viewing the submission.
- The system will allow a rating to be applied by the reviewer, i.e. like, neutral, or dislike.
- The system will allow the reviewer to suggest a time, date, and venue for a showing of a film.

### 2.2.4.3. Processing

- The system will use the calendar package to convert the scheduled time

### 2.2.4.4. Output

- The system will add the reviewer's comments to the Film Submission collection.
- The system will add the reviewer's rating to the Film Submission collection.
- The system will add time/date/venue suggestions to the Film Submission collection.
- The system will update the count of completed reviews for this film in the submission management section.

### 2.2.4.5. Error handling:

- n/a

## 2.2.5. Submissions Management

### 2.2.5.1. Introduction

The application will allow the administrator to see all films submitted by filmmakers, make sure the media files submitted have been formatted properly, and tag films for review by staff members.

### 2.2.5.2. Inputs

- The system will draw inputs from the Film Submission collection, based on the inputs from 2.2.3: Film Submission.

### 2.2.5.3. Processing

- The system aggregates all submission, their metadata, and reviews into tabular format.

### 2.2.5.4. Outputs

- The system will populate a table based on data from the Film Submission collection.

- The system will allow the admin to view the staff's progress on completing reviews.
- The system will allow the admin to note that a film has been fully reviewed and is ready to schedule.

#### 2.2.5.5. *Error handling*

- n/a

### 2.2.6. Festival Schedule Creation

#### 2.2.6.1. *Introduction*

The application will allow the administrator to create a bookings schedule for the films to be shown during the festival. The proposed time slots gathered from all staff reviews will be available as suggestions.

#### 2.2.6.2. *Inputs*

- The system will take in films approved during 2.2.5: Submissions Management, stored in the Film Submission collection.

#### 2.2.6.3. *Processing*

- The system will format a calendar view of the film festival week.
- The system will allow the admin to view the time slots that each reviewer proposed, per film.
- The system will allow the admin to assign each film to a time slot and venue

#### 2.2.6.4. *Outputs*

- The system will create an entry in the Screenings Calendar collection for each distinct film screening time slot and venue created.

#### 2.2.6.5. *Error handling*

- The system will display conflicting time slots at the same venue.

### 2.2.7. Bookings Management

#### 2.2.7.1. *Introduction*

The application will allow the administrator to manage the scheduled films, creating a film details page for the online booking store for each film that has been scheduled, and manage what tickets have been sold.

#### 2.2.7.2. *Inputs*

- The System will draw in inputs from the Screenings Calendar collection, based on the output from 2.2.6: Festival Schedule Creation.
- The system will take in a thumbnail for each film currently available for booking.
- The system will take in a brief description of each film.

- The system will draw in inputs from the Bookings collection, based on the output from 2.2.8: Film Bookings and Payment.

#### 2.2.7.3. *Processing*

- The system will update the bookings page of the application and populate it with the films that were approved for viewing and their details.

#### 2.2.7.4. *Outputs*

- The system stores the thumbnail and description for each film in the Film Submission collection in MongoDB for future access.

#### 2.2.7.5. *Error handling*

- The system will monitor for conflicting scheduling of films.

### 2.2.8. **Film Bookings and Payment**

#### 2.2.8.1. *Introduction*

The application will allow a visitor to the web site to browse the scheduled films and select films they wish to attend. They will be able to pay for those bookings and receive a ticket for the events.

#### 2.2.8.2. *Inputs*

- The system will take the attendee's name.
- The system will take the attendee's email.
- The system will take the films selected for purchase.
- The system will take the film time and date from the Screenings Calendar collection.
- The system will take the attendee's payment details.

#### 2.2.8.3. *Processing*

- The system will put an order in the shopping cart.
- The system will generate a total cost for the booking in the shopping cart.
- The system will generate an order to be confirmed or altered.
- The system will process the payment with payment details input supplied.
- The system will use a payment service to confirm whether the payment method was valid.

#### 2.2.8.4. *Outputs*

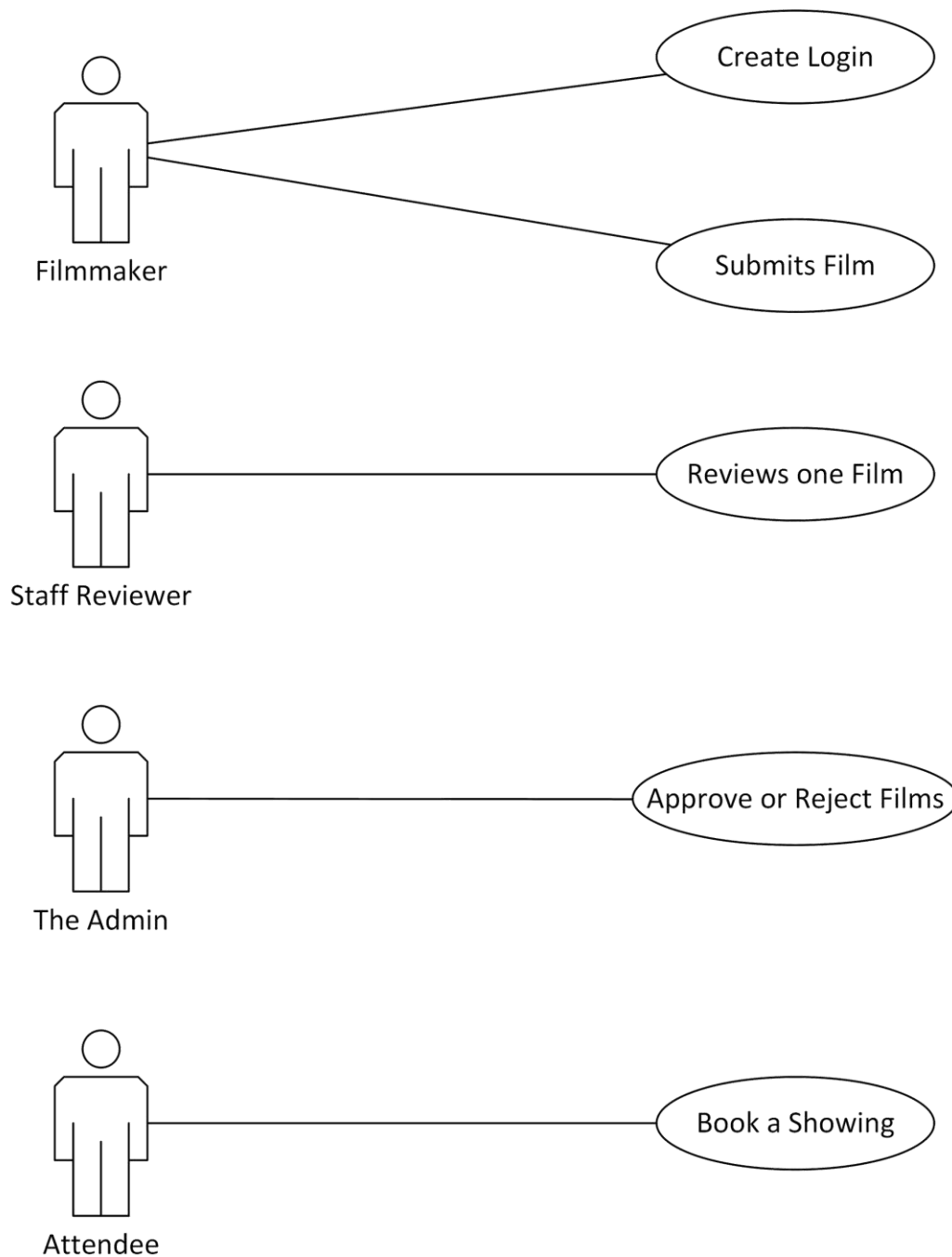
- The system will save booking, if complete.
- The system will email the user confirmation of the order upon successful booking.
- The system will email the user with details if the order was not successful.
- The system will generate a printable ticket with QR code.

#### 2.2.8.5. Error handling

- System will ensure that film screenings selected are still available to book.
- System will reject payment methods if not valid, not letting the purchaser receive tickets.

## 2.3. Use Cases & Use Case Diagrams

### 2.3.1. Use Case Diagrams



### 2.3.2. Use Case Description: Create Login

Use Case Number: 1	Use Case Name: Create Login
Goal: Register a Filmmaker	
Brief Description: A new visitor to the JFF web site who wishes to submit a film to the festival will create a new account.	
Actors: Filmmaker	
Frequency of Execution: Once per user	
Scalability: The estimated number of users submitting films in last year's festival was between 100 and 200.	
Criticality: Essential for submitting films to be judged	
Primary path: The user submits their name, email, contact number, and what filmmaker category they fall into. An account is created for them.	
Alternative Path: The user does not wish to register and does not submit their information. The use case ends without attempting to make an account.	
Exception: The email the user submits is already in the system. The process fails, and an account is not created for them.	

### 2.3.3. Use Case Description: Submit Film

Use Case Number: 2	Use Case Name: Submit Film
Goal: Enter a film into the JFF	
Brief Description: A registered filmmaker inputs the information about their film and uploads a media file	
Actors: Filmmaker	
Frequency of Execution: Once per user, or additional times if admin detects a problem	
Scalability: The estimated number of films submitted in last year's festival was between 100 and 200. This will require bandwidth and storage for the media files.	
Criticality: Essential for submitting films to be judged	

Primary path: The user inputs their film information, including title, genre, film length, description, and comments. They browse for a media file to upload. The film is submitted and stored on the JFF server.

Alternative Path: The admin detects a problem with the media file, and asks the filmmaker to submit a fixed file.

Exception: The submission fails due to network problems. The filmmaker must submit again.

### 2.3.4. Use Case Description: Review one Film

Use Case Number: 3	Use Case Name: Reviews one Film
Goal: One of the JFF staff members watches and reviews one film	
Brief Description: The staff member chooses a submitted film and watches it. They submit comments, choose a rating, and might propose a time slot for the film.	
Actors: Staff Reviewer	
Frequency of Execution: Once per submitted film per staff member	
Scalability: At least two staff members will review each film	
Criticality: Must be completed before films are allowed to be shown at the Junior Film Fleadh	
Primary path: The staff member views their queue of films to review. They choose a film, and watch the film as a streaming media file. They write comments, and note if they approve, deny, or are neutral about the film. If approved or neutral, they will propose a time slot for the film during the festival.	
Alternative Path: The staff member may leave the film's review page without submitting the review. Their input will be saved until later.	
Exception: The media file fails to stream, and the admin must get a fixed media file from the filmmaker.	

### 2.3.5. Use Case Description: Approve or Reject Films

Use Case Number: 4	Use Case Name: Approve or Reject Films
Goal: All films to be shown during the festival are approved and a schedule is finalized.	
Brief Description: The administrator views all film reviews, along with the staff reviews and proposed time slots for showings. They decide on whether to include each film. They arrange the films into a schedule for	



the film festival week.
Actors: The Admin
Frequency of Execution: Once per year; can be revised if needed
Scalability: Only a festival administrator can access
Criticality: Must be completed to have a schedule of film showings
Primary path: The Admin views the list of all films that have been submitted. When examining each one, the staff reviews, comments, and rating will be available. They decide if each film is approved or rejected. They use the suggested time slots from staff to arrange all films into a calendar of screenings for the festival. If the schedule is finalised, it is submitted, and will be sent to the print design firm to design the festival handouts.
Alternative Path: The Admin views the films and reviews, but not enough necessary reviews have been submitted. They must revisit this process once more reviews are in.
Alternative Path: The Admin views the films and reviews, but cannot slot all films into a schedule without input from their staff. They leave the website, the current schedule is saved for them, and they revisit the process later.

### 2.3.6. Use Case Description: Book a Showing

Use Case Number: 5	Use Case Name: Book a Showing
Goal: A potential festival attendee purchases tickets to film showings	
Brief Description: An attendee views the schedule of films and detailed descriptions in the festival. They add films they wish to attend to a cart and then check out and pay for tickets.	
Actors: Attendee	
Frequency of Execution: Every time a person wishes to buy a ticket	
Scalability: Must be able to handle the volume of ticket sales. Current estimates for ticket sales show that they spike two to three days before the festival, with analytics showing 10-15 simultaneous users booking at peak.	
Criticality: Must be completed to sell tickets to the festival. If this fails, then a potential attendee can only get tickets by calling the festival office.	

Primary path: An attendee views the calendar, which shows the film's title and show times along with a thumbnail. They click on a film to view details, which includes a description, the filmmaker's name, and a larger picture or a streaming trailer. If they wish to attend this film, they add it to their cart. When satisfied, they view their cart, and check out. They receive an email with their ticket and a receipt.

Alternative Path: The attendee views the film calendar and details, but they do not wish to attend any films. They leave the web site.

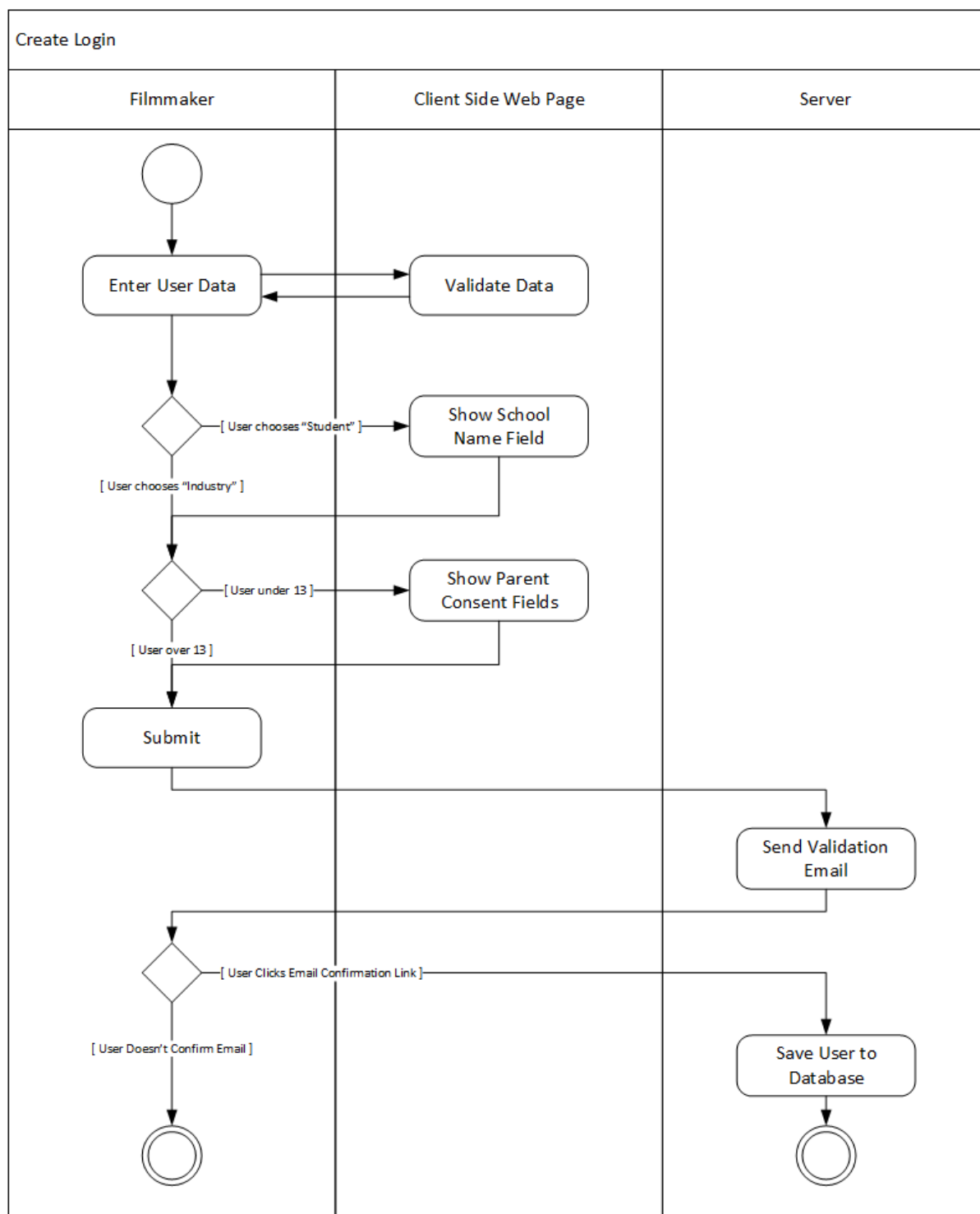
Alternative Path: The attendee views the film calendar and details, but the films they are interested in are booked out. They leave the web site.

Exception: The attendee adds films to their cart, and starts the checkout process, but the server detects one of the films they wish to attend has been sold out in the meantime. The checkout process ends before their payment is processed, and the error is displayed. They have a chance to edit their cart again.

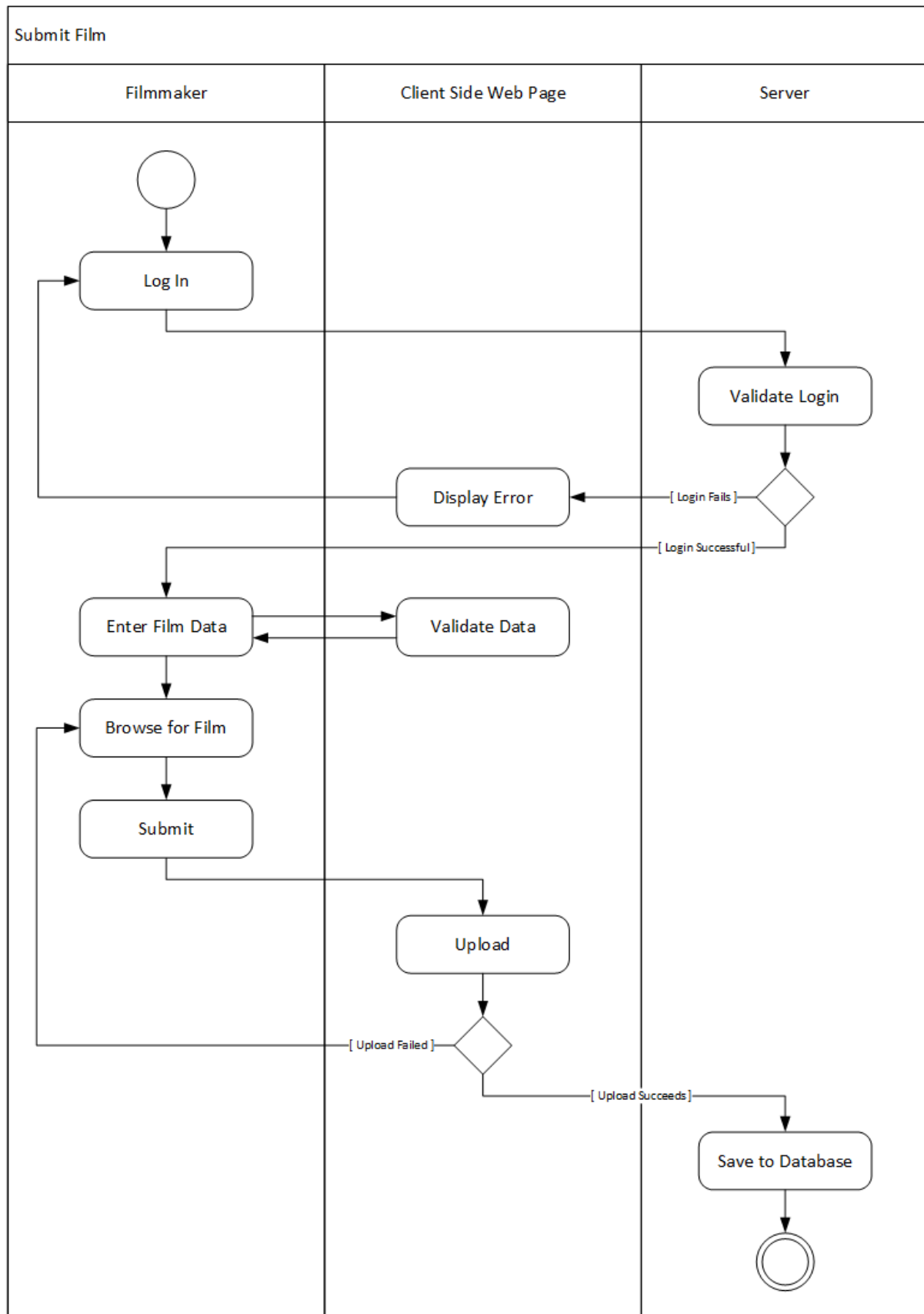
Exception: The attendee checks out, but their payment does not process successfully. They are emailed a failed receipt from the payment processing service. They have a chance to edit their cart again.

## 2.4. Activity Diagrams

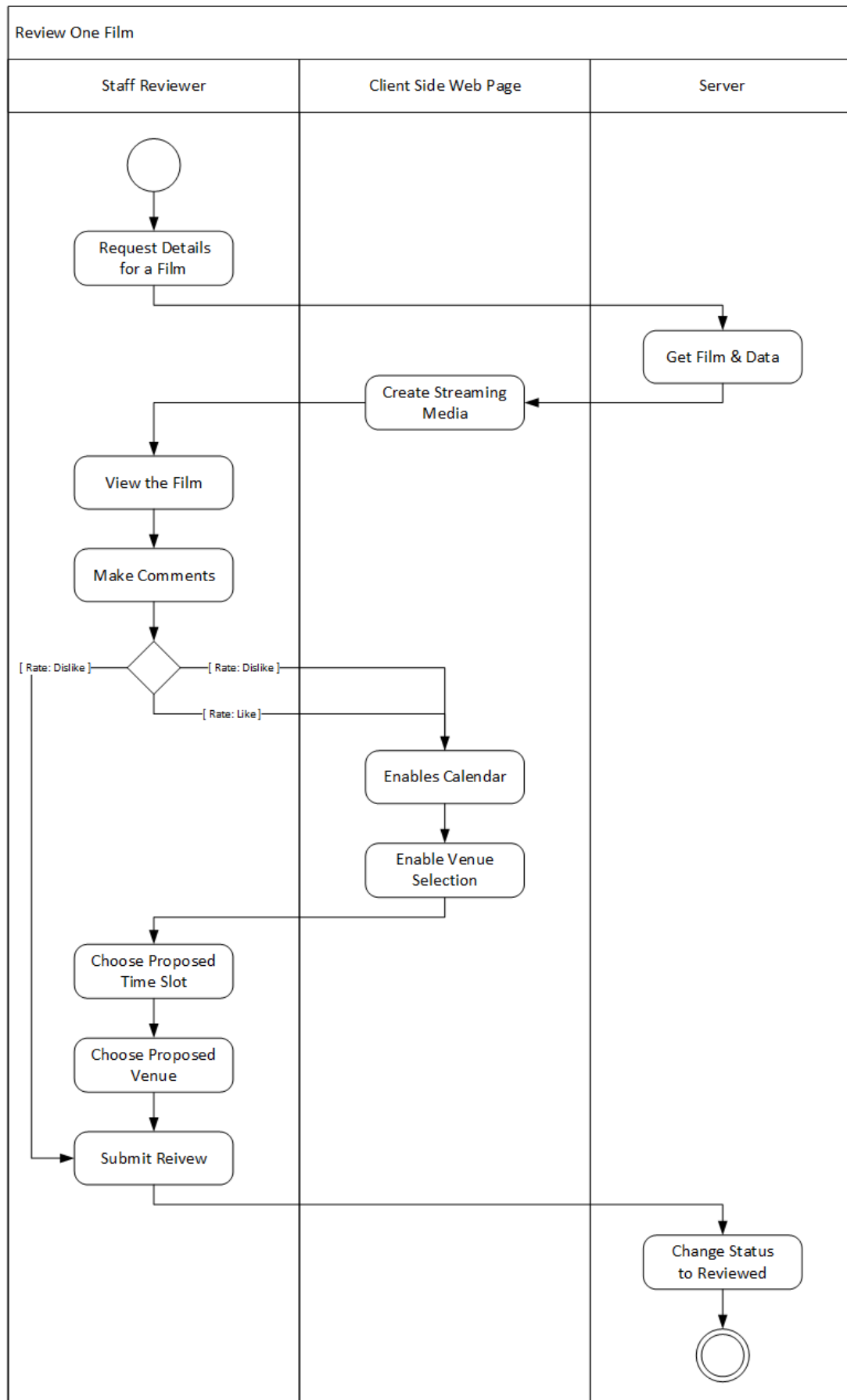
### 2.4.1. Activity Diagram for Use Case #1



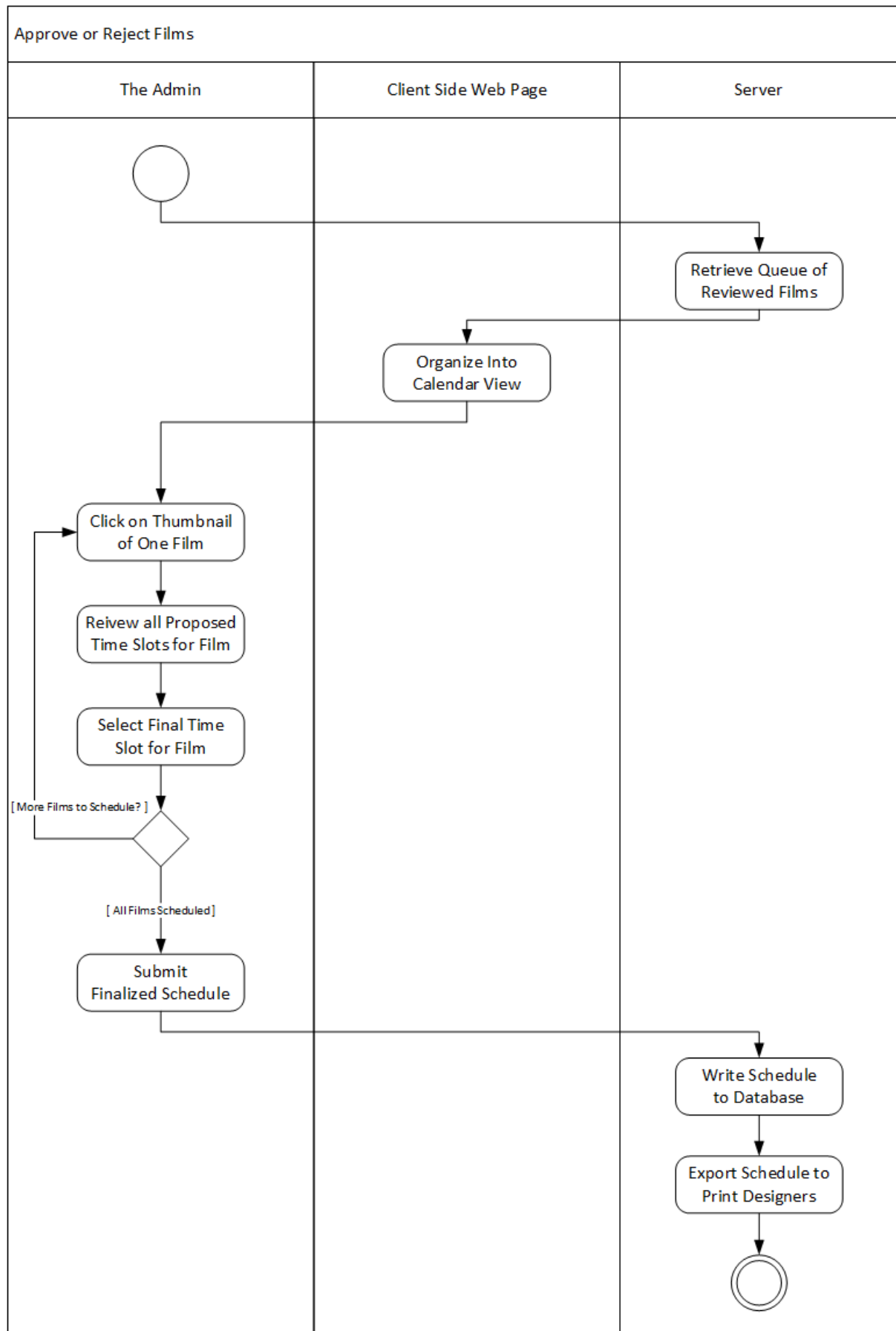
### 2.4.2. Activity Diagram for Use Case #2



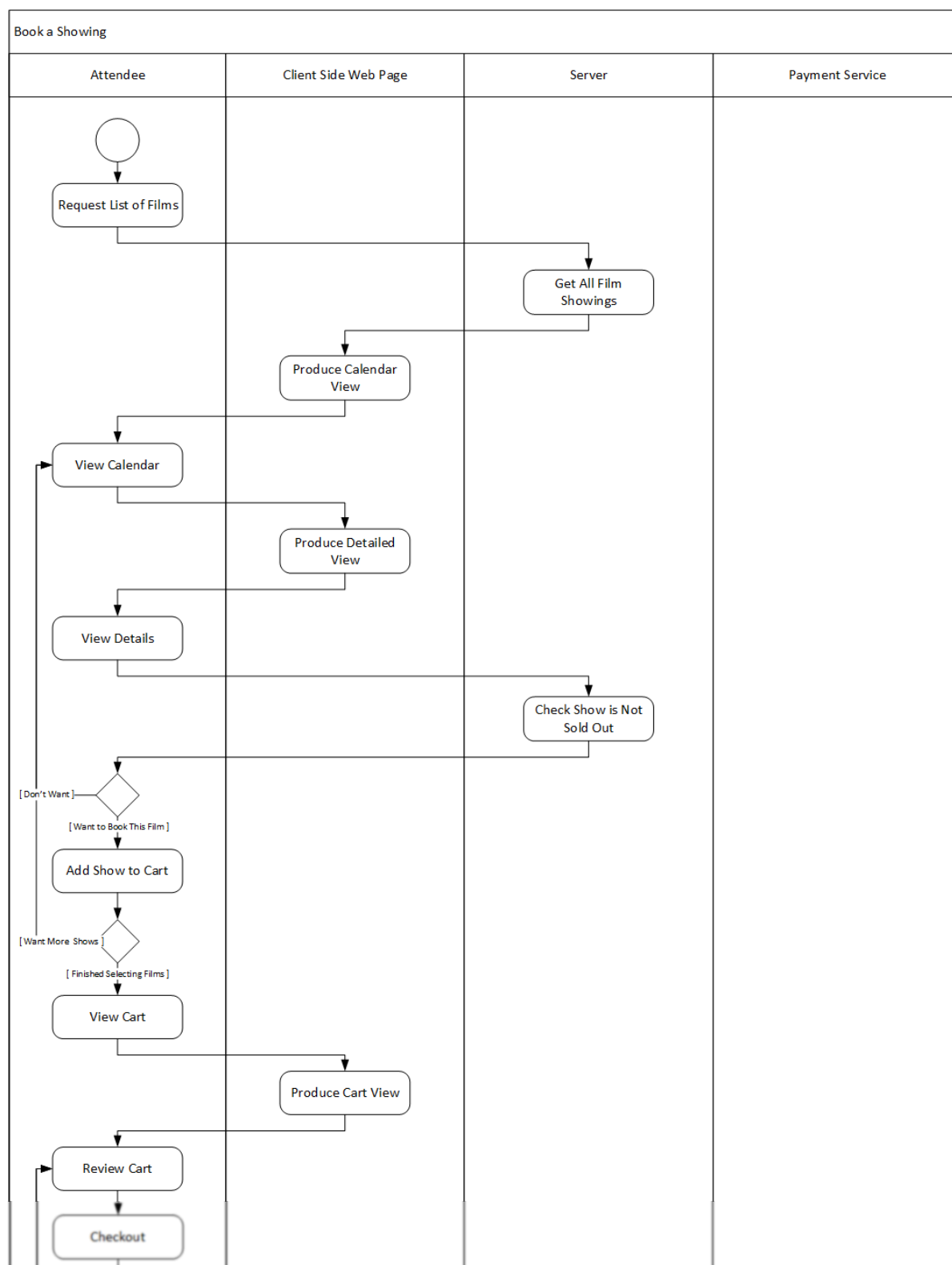
### 2.4.3. Activity Diagram for Use Case #3



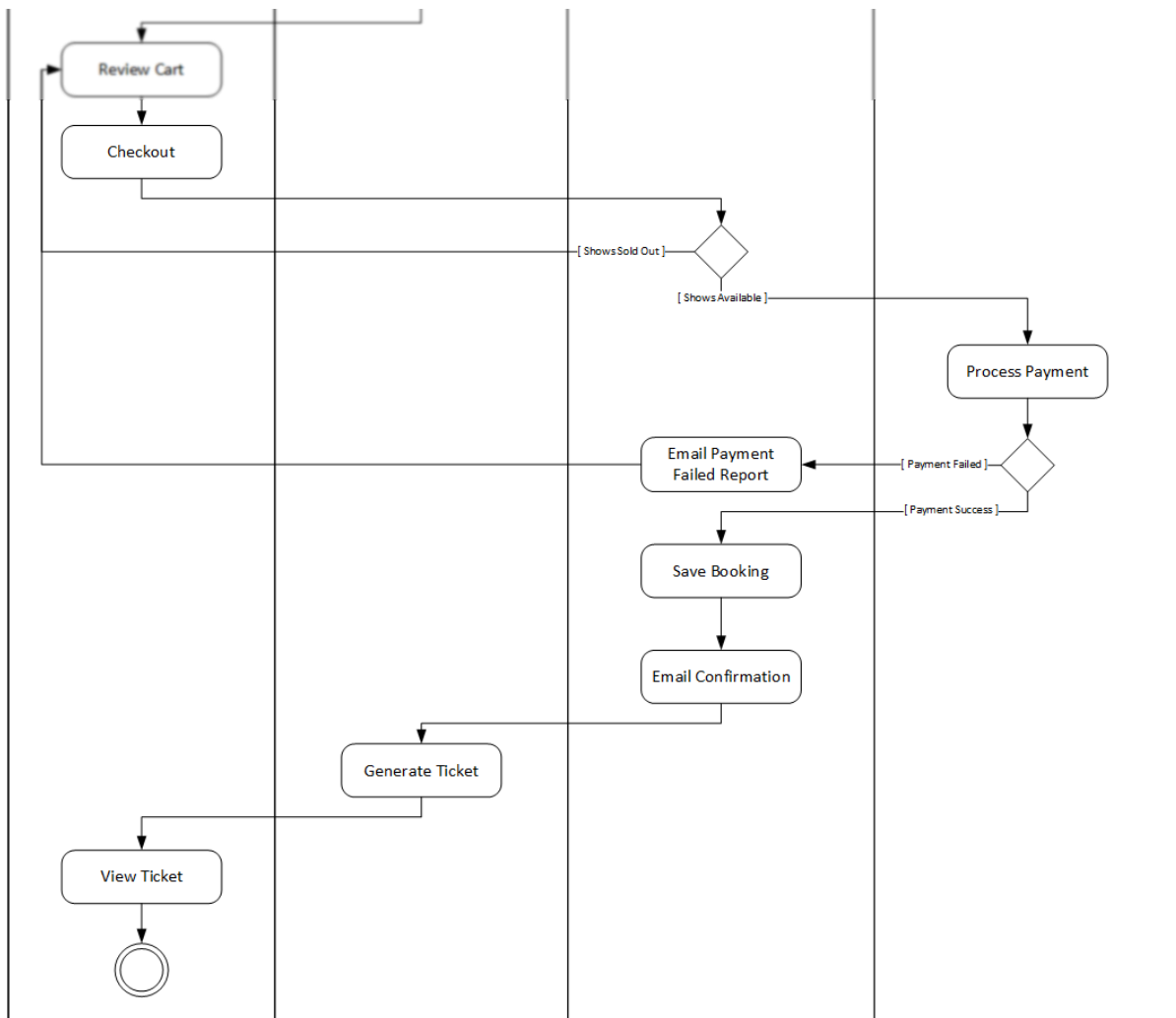
### 2.4.4. Activity Diagram for Use Case #4



## 2.4.5. Activity Diagram for Use Case #5



*Diagram continued*

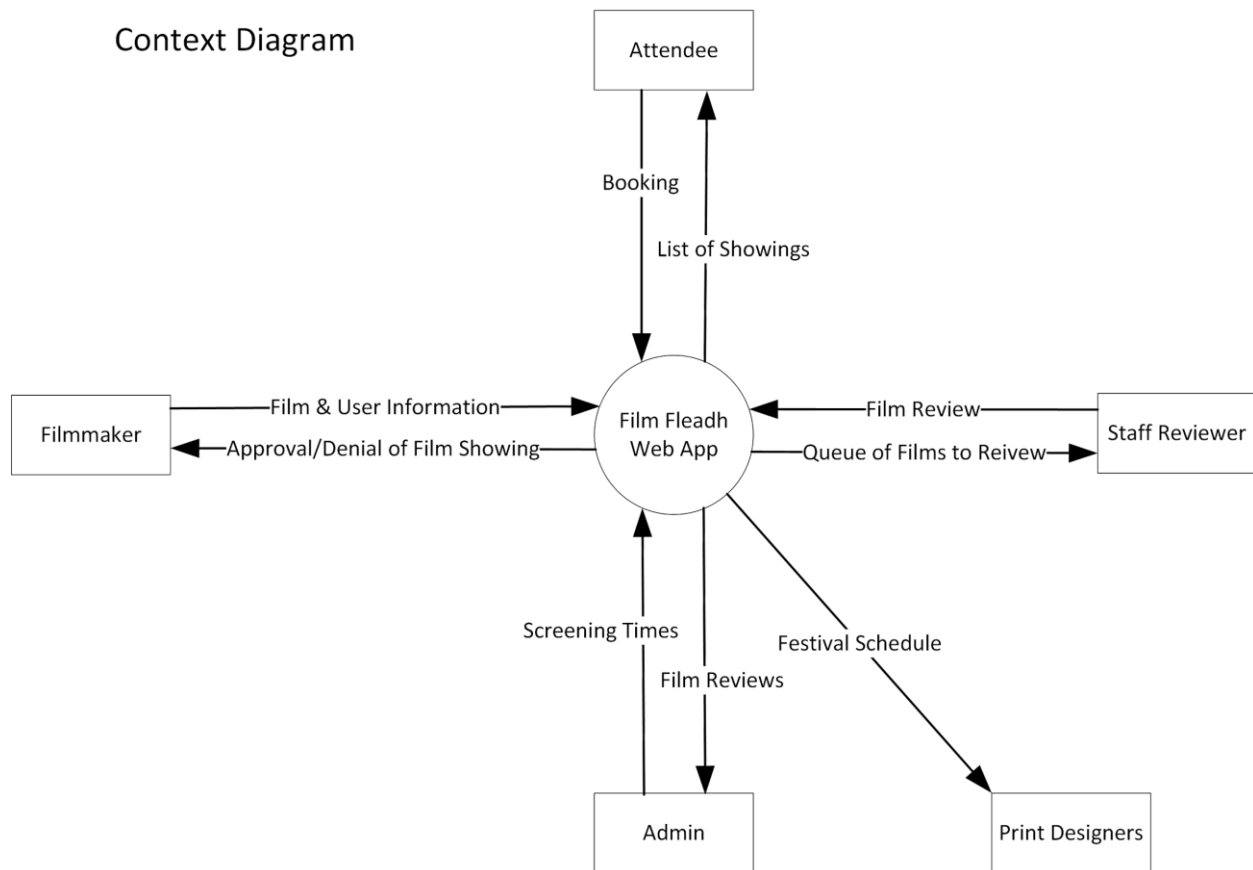




## 2.5. Data Flow Diagrams

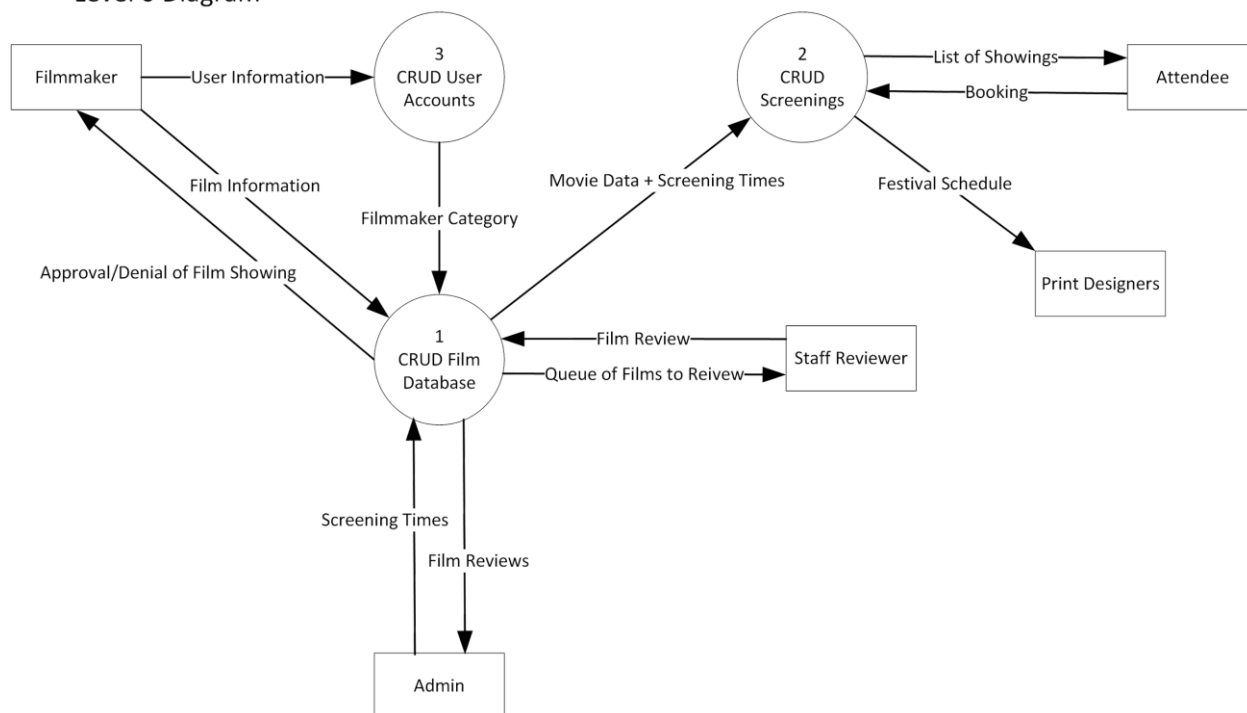
### 2.5.1. Context Diagram

Context Diagram



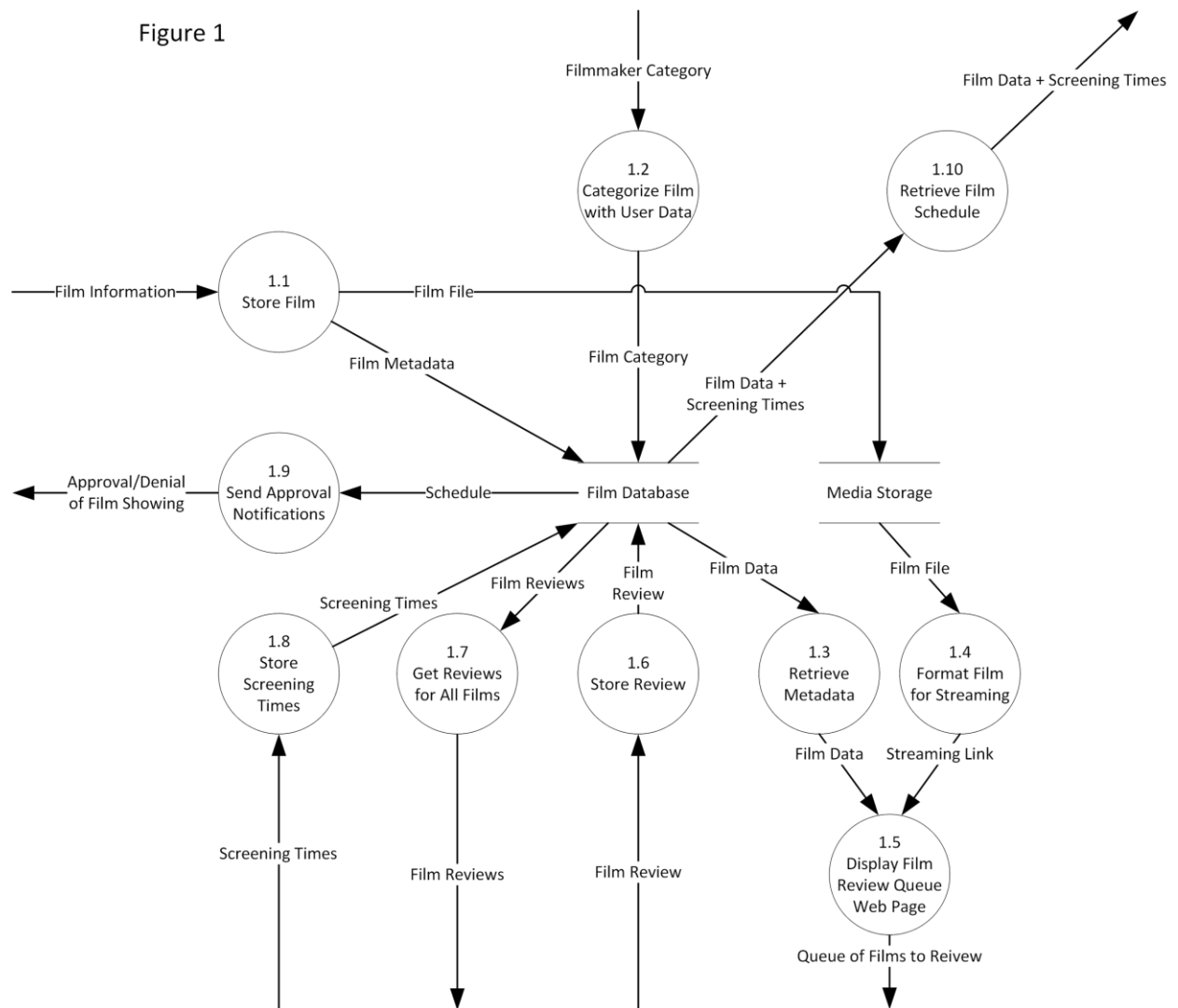
## 2.5.2. Data Flow Diagram: Level 0 Diagram

Level 0 Diagram



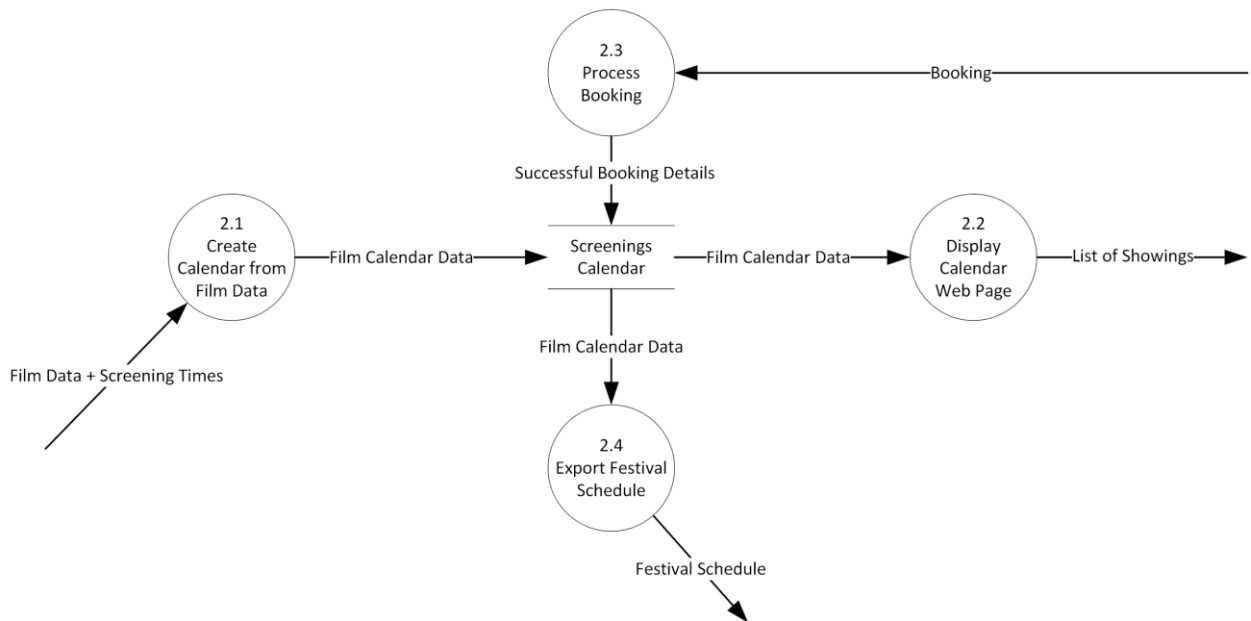
### 2.5.3. Data Flow Diagram: Figure 1

Figure 1



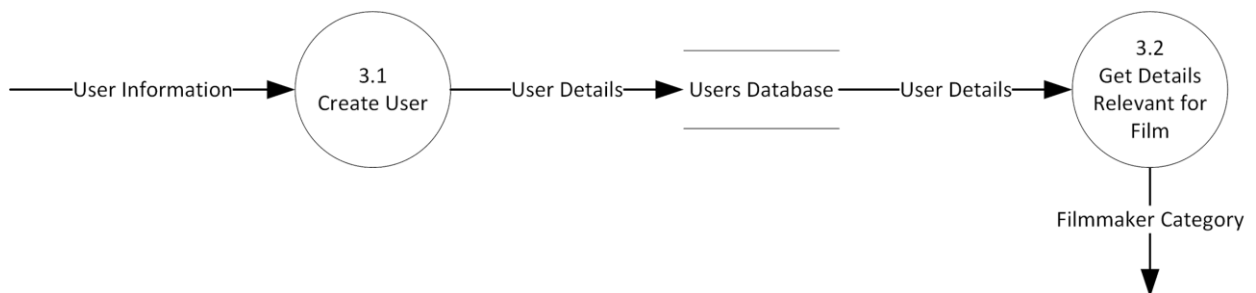
### 2.5.4. Data Flow Diagram: Figure 2

Figure 2



### 2.5.5. Data Flow Diagram: Figure 3

Figure 3



## 2.6. Database Schema (JSON)

### 2.6.1. Film Submission Collection

```
{ //film submission collection
  "title" : "film1",
  "genre" : "thriller",
  "length": "feature",
  "shortDescript": "bla bla",
  "addComm" : "bla bla bla",
  "reviewComm" : "bla bla"
  "storageLink" : "http...",
  "suggestTime" : "00:00",
  "suggestDate" : "dd/mm/yy",
  "suggestVenue" : "place"
}
```

### 2.6.2. Screening Calendar Collection

```
{ //screening cal
  "shortDescript": "bla bla",
  "date" : "dd/mm/yy",
  "time" : "00:00",
  "venue" : "place"
}
```

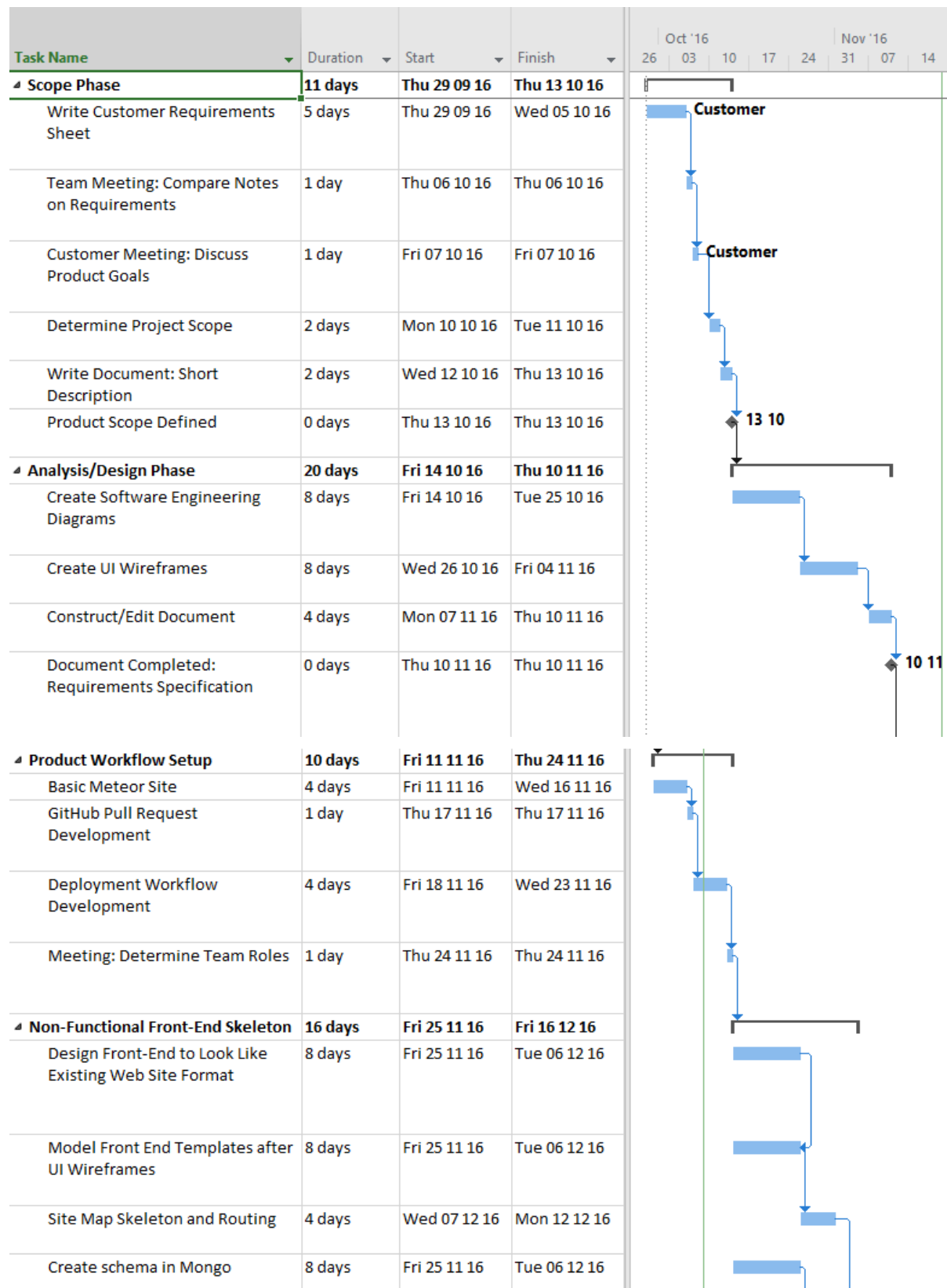
### 2.6.3. Users Collection

```
{//user creation
  "name" : "david",
  "email" : "test@test.com",
  "contactNo" : "12345678",
  "userType"{
    "teacher" : "bool",
    "student" : "bool",
    "other" : "info",
  }
  "terms" : "bool"
}
```

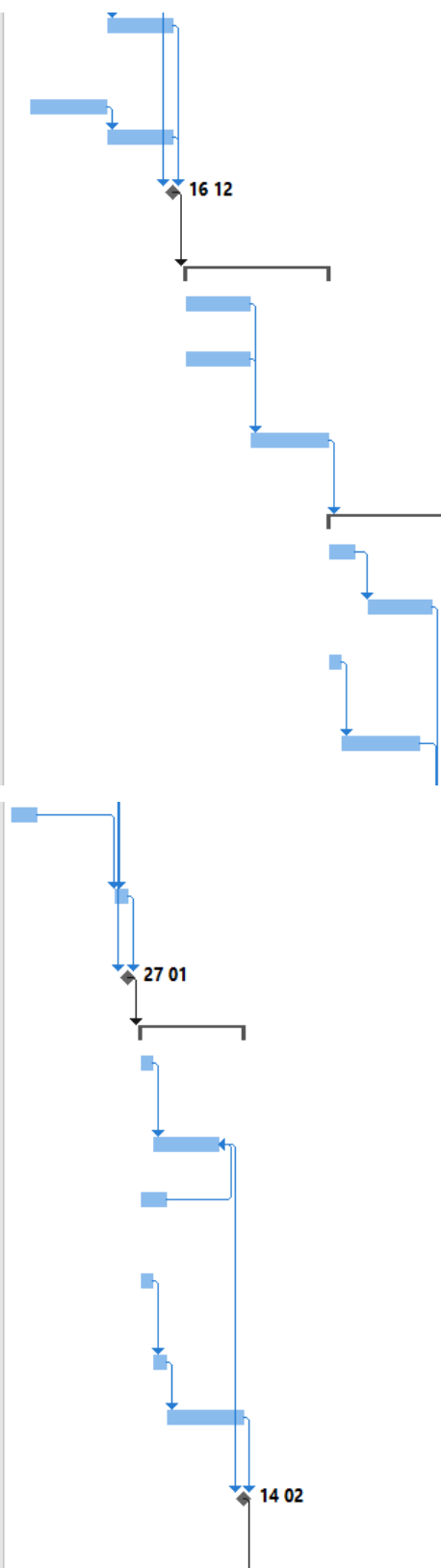
#### 2.6.4. Bookings Collection

```
{//bookings available collection
  "title" : "film1",
  "genre" : "thriller",
  "length": "feature",
  "shortDescript": "bla bla",
  "addComm" : "bla bla bla",
  "storageLink" : "http..."
  "price" : "€.." //place upload button on submission UI
}
```

### 3. Project Timeline

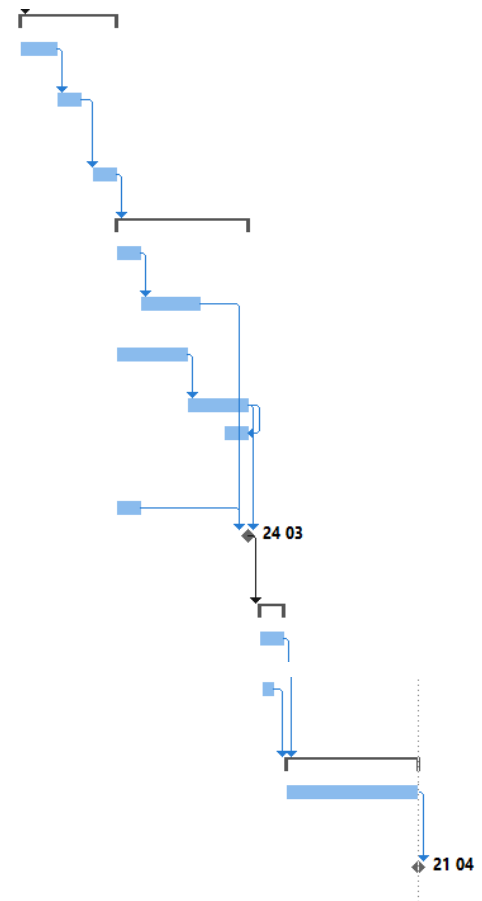


Create Meteor Templates to Show Data	8 days	Wed 07 12 16	Fri 16 12 16
Login System	8 days	Fri 25 11 16	Tue 06 12 16
Admin/User Security	8 days	Wed 07 12 16	Fri 16 12 16
Functional Site (No Business Logic)	0 days	Fri 16 12 16	Fri 16 12 16
<b>Feature: User Creation</b>	<b>16 days</b>	<b>Mon 19 12 16</b>	<b>Mon 09 01 17</b>
Create New User Form	8 days	Mon 19 12 16	Wed 28 12 16
Add Users to Database Using Meteor	8 days	Mon 19 12 16	Wed 28 12 16
Integrate Front End and Back End	8 days	Thu 29 12 16	Mon 09 01 17
<b>Feature: Film Submission</b>	<b>14 days</b>	<b>Tue 10 01 17</b>	<b>Fri 27 01 17</b>
Create User Upload Form	4 days	Tue 10 01 17	Fri 13 01 17
Backend Code to Save to Database	8 days	Mon 16 01 17	Wed 25 01 17
Find Media Upload Meteor Package	2 days	Tue 10 01 17	Wed 11 01 17
Use Package to Save Uploaded Media	8 days	Thu 12 01 17	Mon 23 01 17
Mail Server & Notifications Integration	4 days	Tue 10 01 17	Fri 13 01 17
Link Email Admin to Film Submission	2 days	Thu 26 01 17	Fri 27 01 17
Film Can Be Uploaded	0 days	Fri 27 01 17	Fri 27 01 17
<b>Feature: Film Reviews</b>	<b>12 days</b>	<b>Mon 30 01 17</b>	<b>Tue 14 02 17</b>
Update Database to Have User Roles	2 days	Mon 30 01 17	Tue 31 01 17
Create UI For Staff Film Reviews	8 days	Wed 01 02 17	Fri 10 02 17
Use Media Package to Display Streaming	4 days	Mon 30 01 17	Thu 02 02 17
Create Logic To Save Reviews to DB	2 days	Mon 30 01 17	Tue 31 01 17
Find Calendar Meteor Package	2 days	Wed 01 02 17	Thu 02 02 17
Integrate Calendar Package with Scheduling	8 days	Fri 03 02 17	Tue 14 02 17
Customer Meeting: Show Submission and Review MVP	0 days	Tue 14 02 17	Tue 14 02 17





<b>Feature: Calendar Management</b>	<b>12 days</b>	<b>Wed 15 02 17</b>	<b>Thu 02 03 17</b>
Create UI for Admin View Films	4 days	Wed 15 02 17	Mon 20 02 17
Integrate Calendar on Films Page	4 days	Tue 21 02 17	Fri 24 02 17
Write Schedule to Database	4 days	Mon 27 02 17	Thu 02 03 17
<b>Feature: Bookings</b>	<b>16 days</b>	<b>Fri 03 03 17</b>	<b>Fri 24 03 17</b>
Retrieve All Films from DB	2 days	Fri 03 03 17	Mon 06 03 17
Create Bookings Listing Page	8 days	Tue 07 03 17	Thu 16 03 17
Create Shopping Cart	8 days	Fri 03 03 17	Tue 14 03 17
Create Checkout	8 days	Wed 15 03 17	Fri 24 03 17
Integrate With Payment Processing	4 days	Tue 21 03 17	Fri 24 03 17
Email Receipt	2 days	Fri 03 03 17	Mon 06 03 17
Customer Meeting: Show Bookings MVP	0 days	Fri 24 03 17	Fri 24 03 17
<b>Feature: User Management</b>	<b>4 days</b>	<b>Mon 27 03 17</b>	<b>Thu 30 03 17</b>
Create User View Page	4 days	Mon 27 03 17	Thu 30 03 17
Allow Admin to Define User Roles in DB	2 days	Mon 27 03 17	Tue 28 03 17
<b>Final Testing Phase</b>	<b>16 days</b>	<b>Fri 31 03 17</b>	<b>Fri 21 04 17</b>
Run Through Testing with Team	16 days	Fri 31 03 17	Fri 21 04 17
Provide Application to Client for Testing	0 days	Fri 21 04 17	Fri 21 04 17



## A. Appendices

### A.1 Appendix 1 – Link to GitHub Repository

<https://github.com/davedodea/ProjectX/>

Please contact Dave O'Dea ([d.odea4@nuigalway.ie](mailto:d.odea4@nuigalway.ie)) for access to private repo.

### A.2. Appendix 2 – Short Requirements Document

## Film Festival Web App

Team Try Catch – Andrew East, Chika Onyia, David O'Dea, James Quaife

### Project Goal

Develop a web application for the Junior Film Fleadh in Galway

### Functionality

- Film creators will submit their films to be entered into the festival
- Students will submit written scripts for judgement
- Festival officials will judge films and scripts; each judge will record notes and film rankings
- Festival events will be arranged into a schedule
- Potential festival attendees will view film schedule and book tickets
- Attendees will view workshop schedule and book tickets
- Post-festival, schools can view and book available outreach programmes
- Schedule and meta data can be exported from database to send to graphic design firm to create festival program

### Minimum Viable Product

The minimum functionality to be a success to our client is to implement the film submission and festival workshop booking portions of the product

### Existing Web Design

<http://www.galwayfilmfleadh.com/index.php/juniorfilmfleadh/>

