# Functional specification
# Movie Rental iPhone app

## May 2012

### Contact details

| Prepared for | | Prepared by | |
|---|---|---|---|
| Contact name | David Burrows, Andy Stopford and Lisa Jones | Contact name | Omer Janjua |
| Job title | N/A | Job title | Software Engineer |
| Company name | Creative Lynx | Company name | Creative Lynx |
| Email | N/A | Email | omer.janjua@creativelynx.co.uk |

### Version control

| Date | Version | Notes |
|---|---|---|
| 02/05/2012 | 1 | Initial draft |
| 09/05/2012 | 2 | Final draft |

# Contents

# Document outline

The aim of this document is to clarify and define all functional requirements and development elements of our proposed solution for the Movie Rental iPhone app.

## Intended audience and reading suggestions

This document is intended to outline functionality and features. This document is intended to be read by technical personnel as well as clearly understood by all (non-technical) parties involved in the project.

# Programme overview

The app is intended to support a fully functional movie rental system. To allow the user to track the stock level of a movie and cross reference with which customer has rented the movie.

## Movie Rental Task

The app must have the following functionality:

Each one of these bullet points will be answered at the end of the document explaining how the task was achieved.

- Customers can be added
- Customers can cancel their account if they have returned all movies
- Customers can reopen their account
- A customer can rent and return movies
- A customer has a rental limit of 3 movies
- Each movie has a stock count
- Movies can be added to the system
- Movies can be removed if all copies have been returned
- Movies have cover art

## Task 1 (3 days)

Write a specification for the above with the following:

- A database design detailing the tables and relationships
- A general description of how the application will work
- A written description of each screen in the application and its desired functionality (including any validation)
- Class design indicating what responsibility each class has

In planning this application, maintain a record of questions asked to other developers, and other resources used.

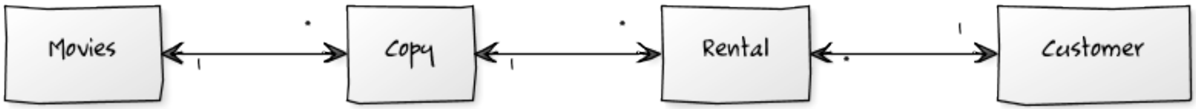
## Testing

When the app is ready for review, we ensure full testing to be done across iOS v5.1.

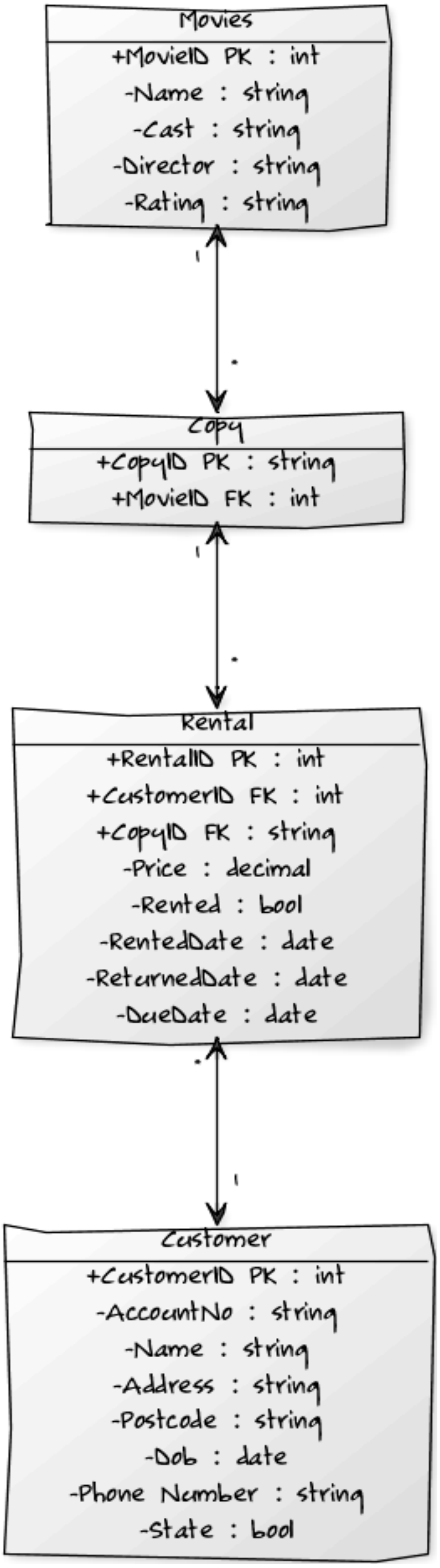
# Concentrating on:

- Detailed commenting
- Apply SOLID Principles
- Using correct design patterns
- Using Dave Verwer’s notes
- Making sure no inconsistency between screens
- If I get stuck take a step back and looking at the problem as a whole. Look at what’s around it and how it works.
- If one method is changed. How will it affect the rest of the app.
- Using OOP principles
- Thoroughly testing the whole app.

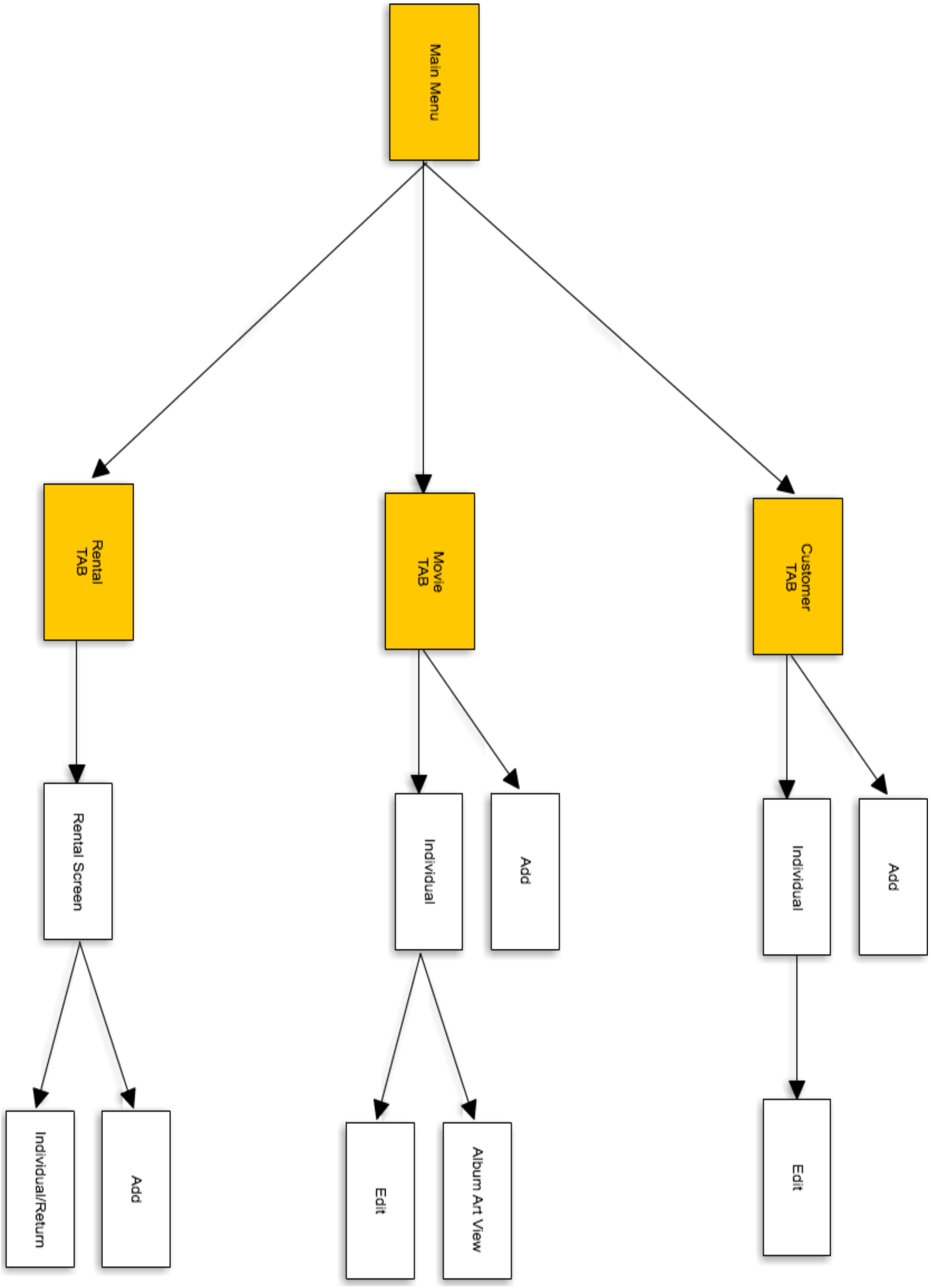
# Class Diagram



# Database Model



# App Screens

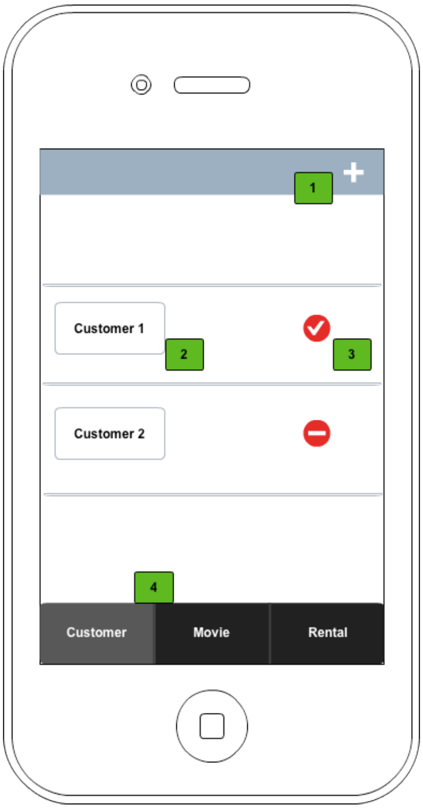

Main Menu
Rental TAB
Movie TAB
Customer TAB
Rental Screen
Individual
Add
Individual
Add
Individual/Return
Add
Edit
Album Art View
Edit

# General App Description:

- The app has 3 tabs. Customer, movie and rental tab.
- Tab1 is the customer tab. This tab allows new customers to be registered. Customer details can be edited. Customers can also get deleted from the system.
- Tab2 allows new movies to be added to the system. Movie information can be edited and deleted if all movies are back in stock.
- Tab3 allows the registered customers to rent movies (up to 3) and return them.

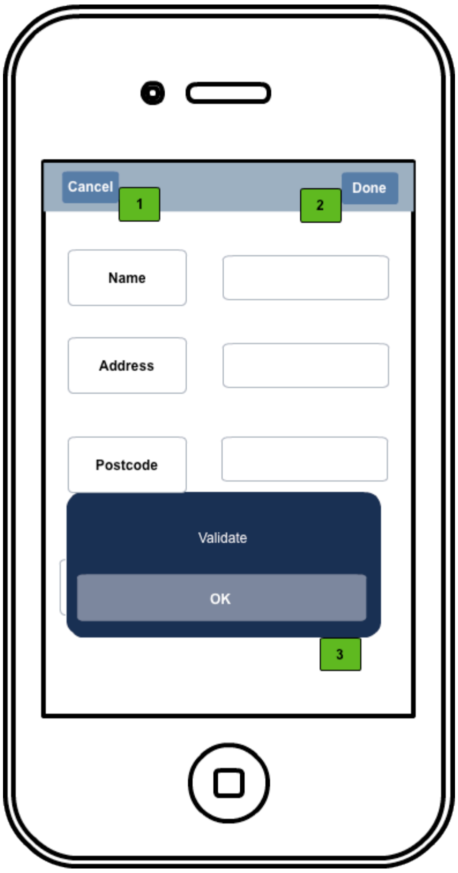
# App Description

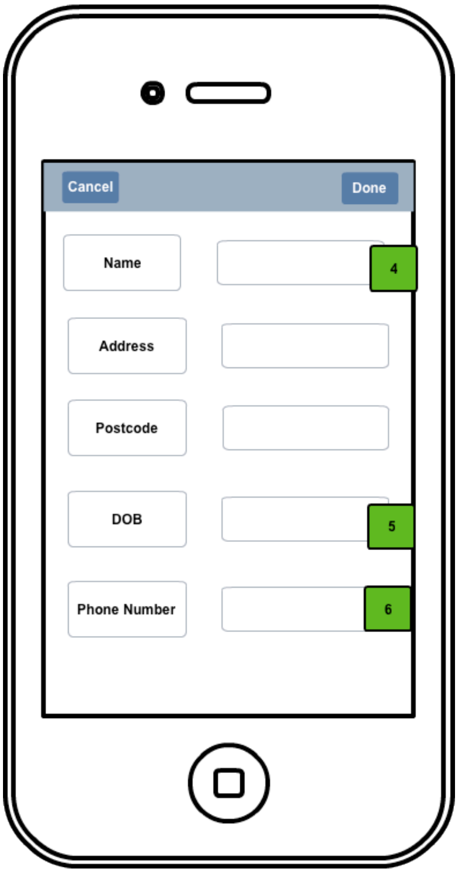
## Customer Screen (Tab 1)



1. This is the add customer button, which navigates the user to the “Customer Add Screen”.
2. This label displays the customer name. Clicking in that area will navigate the user to “Individual Customer Screen”.
3. Thid label shows the customer account state. Weather its open or close.
4. The tabs allow the user to switch between different screens.

# Customer Add Screen

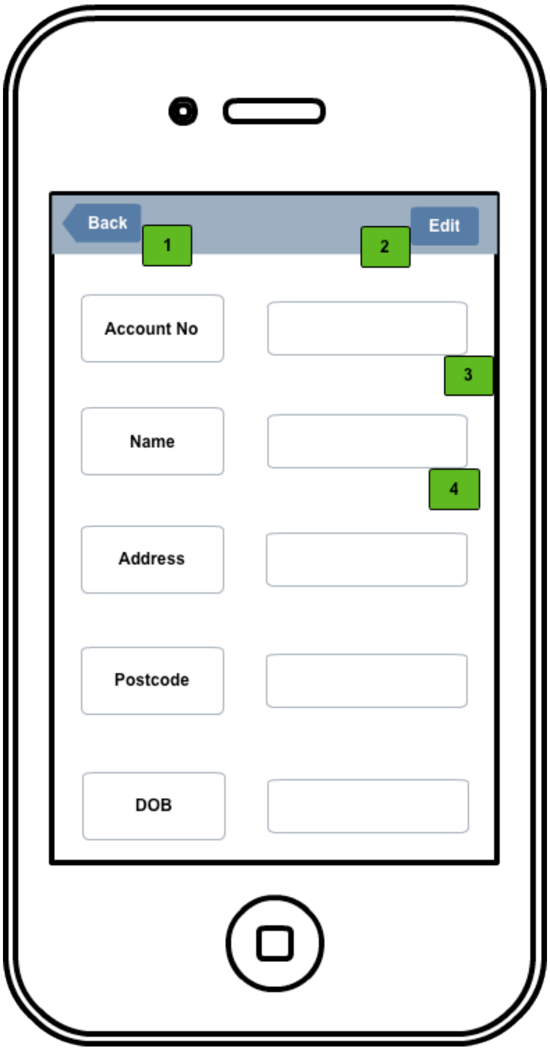




1. Pressing this button will dismiss the screen, discarding any data added.
2. Pressing this button will save the data to the model.
3. A pop up appears if the customer doesn't enter a name, address and postcode.
4. A text field is used to enter the customer name. A standard keyboard will appear in order to enter text into the field.
   -Same action is carried out for the address and the postcode field.
5. A date picker will appear to choose the date of birth.
6. A number keypad pops up in order to enter phone number.

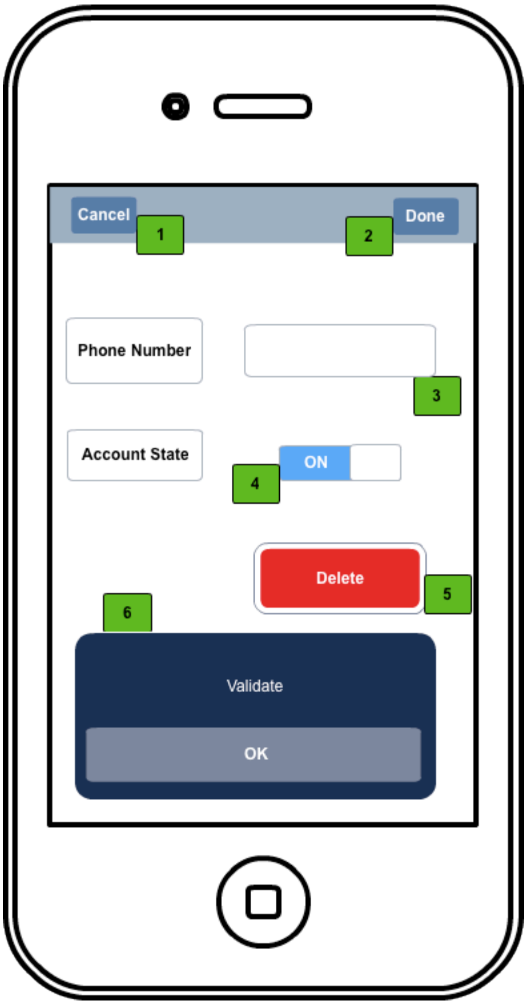
- The account number field is not displayed on the add customer screen since the account number is auto generated.
- Also the account state does appear on the add customer screen since the initial account state is open.

# Individual Customer Screen



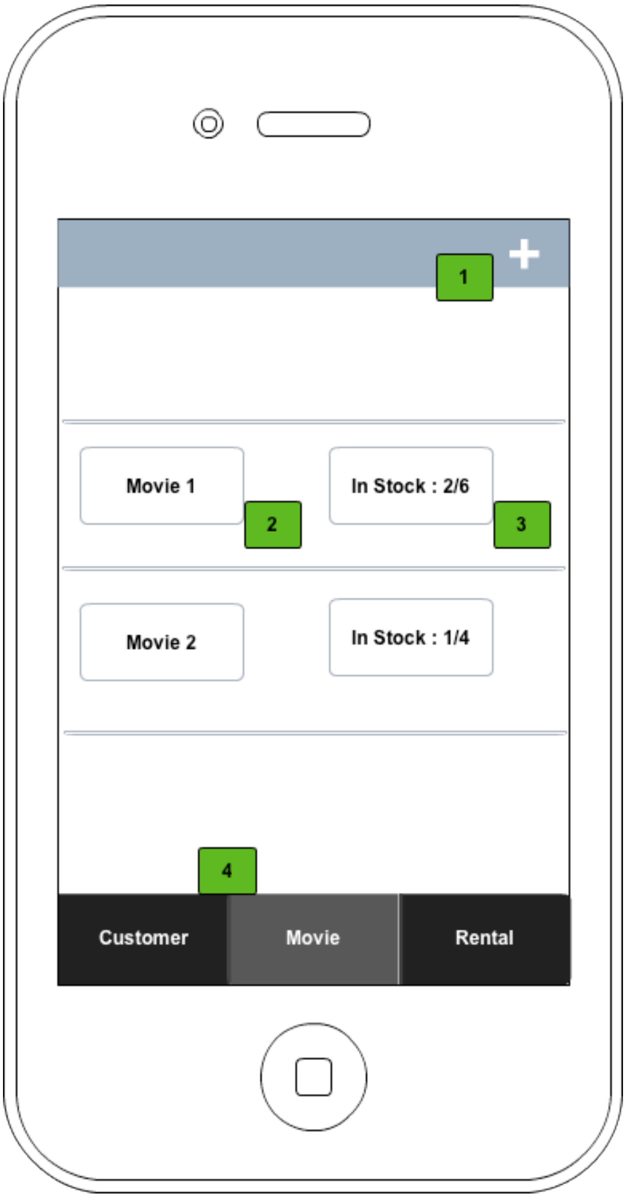
1. Pressing this button will navigate the user back to the main customer screen.
2. Pressing this button will allow the user to edit the information on the screen by modally popping up the Edit Customer Screen.
3. This label displays the auto generated account number.
4. This label displays the relevant information such as the name. Same action is carried out for the following labels.

# Edit Customer Screen



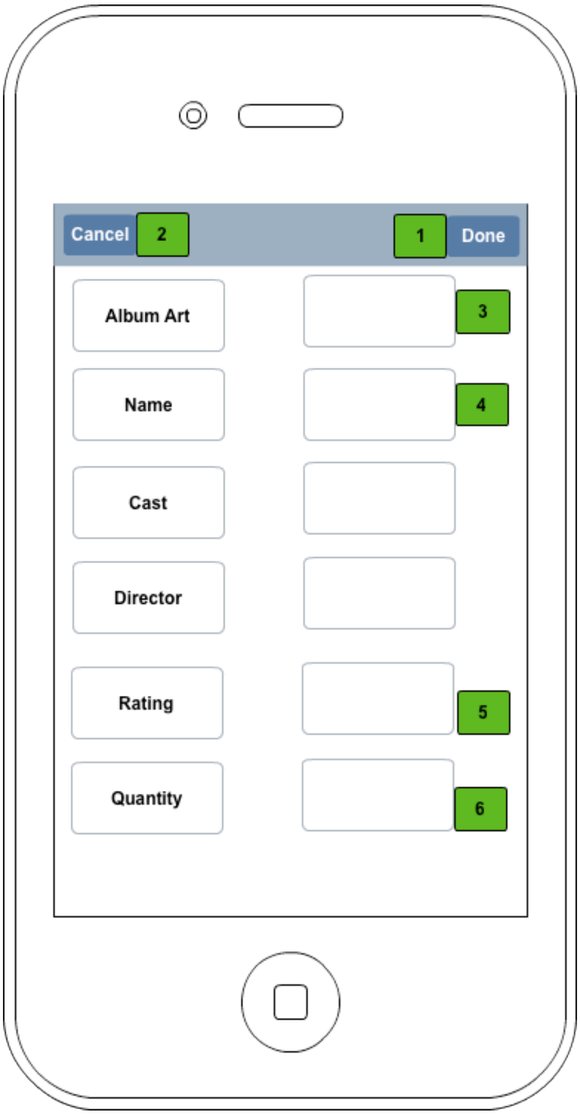
1. Pressing this button will navigate the user back to the Individual Customer Screen.
2. Pressing this button will save any information edited while on the screen.
3. This text field allows the user to edit the information previously added in the Add Customer Screen.
4. This switch is used to change the account state. The account can only be closed if there are no outstanding rentals.
5. By pressing this button account will be permanently deleted. The account can only be deleted if there are no outstanding rentals. This action is non-reversible.
6. Validation pop up appears if the user tries to delete the account with a movie rented out.

# Movie Screen (Tab 2)



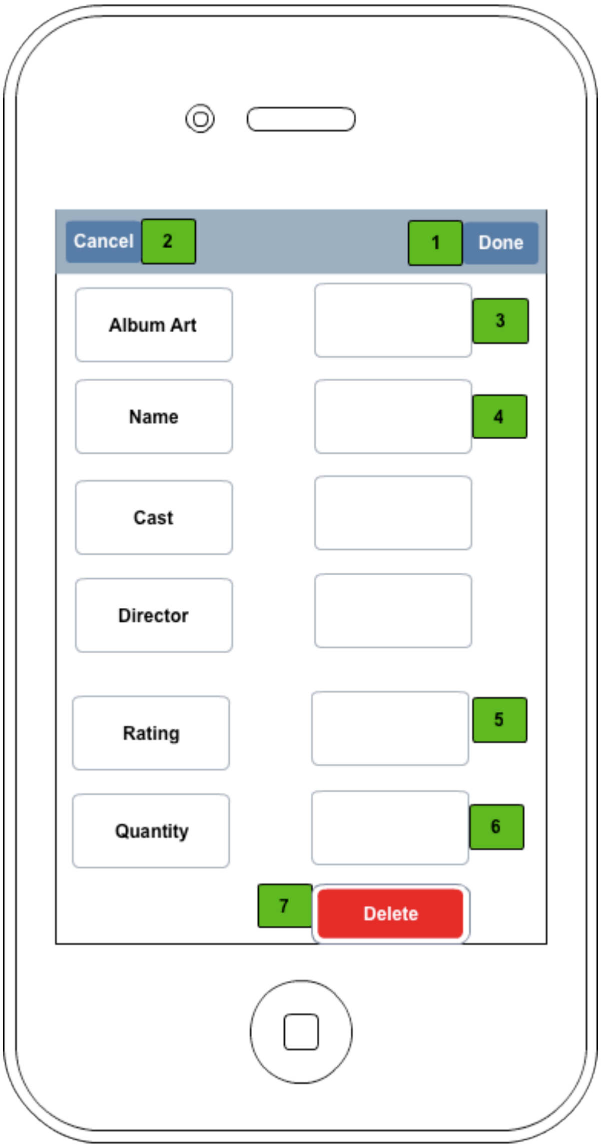
1. Pressing this button will navigate the user to “Add Movie Screen”.
2. Label displaying the movie name. Pressing this area will navigate the user to “Individual Movie Screen”.
3. This label displays the total stock count of the movie against what’s rented out.
4. Tab bar allowing the user to switch between different tabs.

# Add Movie Screen



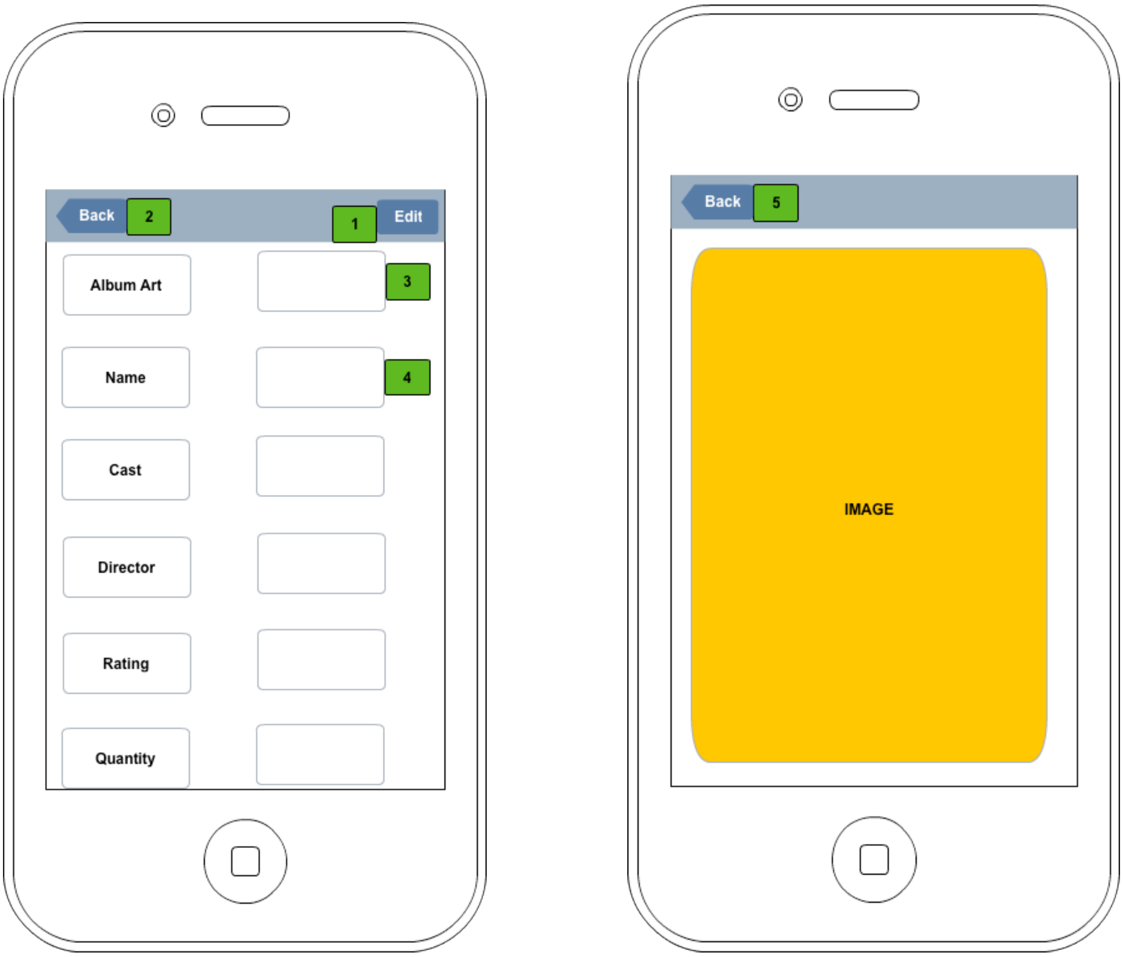
1. Pressing this button will save all the information into the model. Validation is carried to check if a name of the movie and quantity has been entered.
2. Pressing this button will discard any data input.
3. Pressing this button takes the user to the phones photo album to select a picture for the album art.
4. This text field is used to enter in the relevant data. A standard keyboard is used to carry out this action. Same follows for the cast and director.
5. This displays a picker view which allows the user to chose a movie rating e.g. U, PG, 12, 15, 18.
6. Picker view used to enter the quantity of movies the user wants to enter. The movie copy number will be auto generated.

# Edit Movie Screen



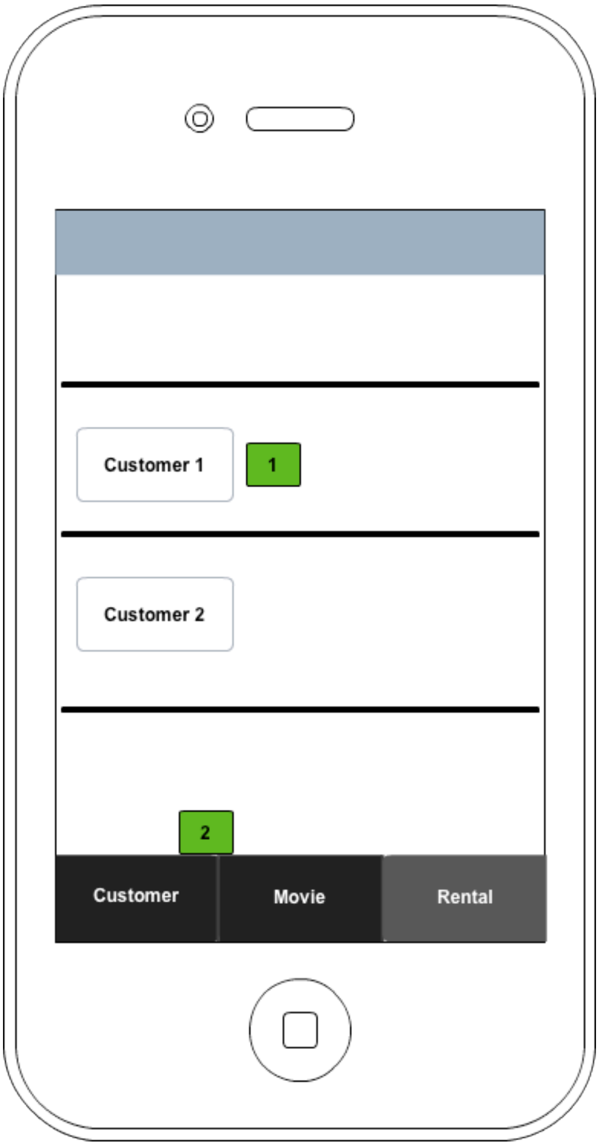
1. Pressing this button will save all the information into the model.
2. Pressing this button will discard any data input.
3. Pressing this button takes the user to the phones photo album to select a picture for the album art.
4. This text field is used to enter in the relevant data. A standard keyboard is used to carry out this action. Same follows for the rest of the fields.
5. The picker view can be used to change the rating of the movie.
6. Picker view used to change the quantity of movies the user wants to enter. The movie copy number will be auto generated.
7. Delete Button used in order to delete the movie if all copies have been returned. If the validation is not met an alert view appears on the screen.

# Individual Movie Screen



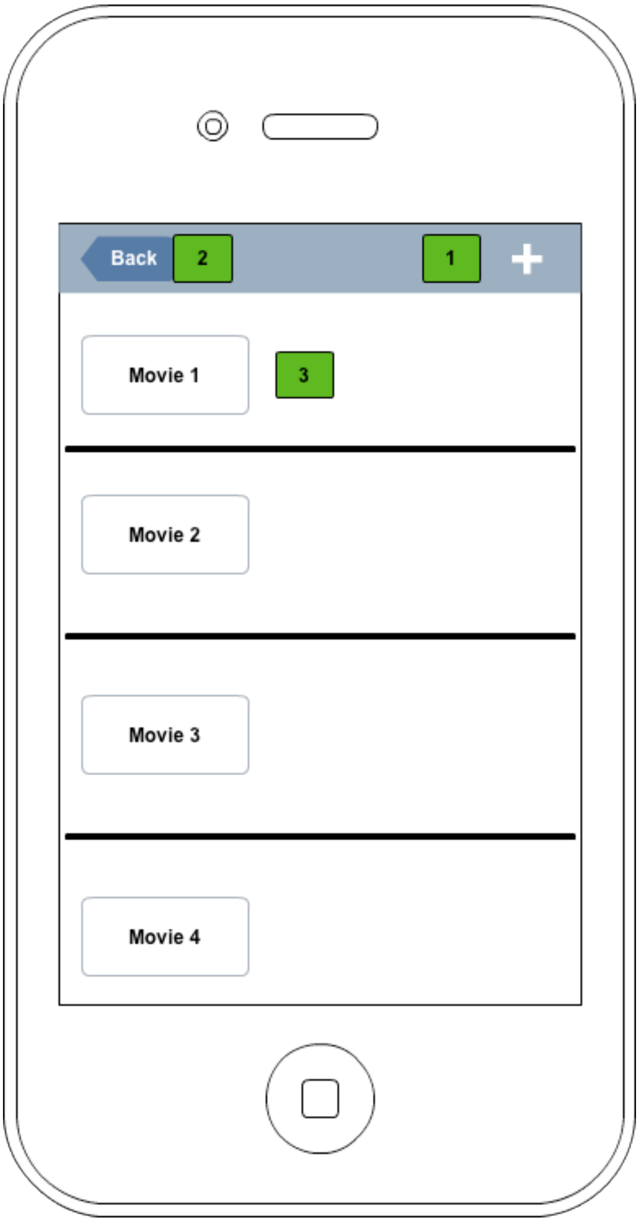
1. User can edit the movie information by pressing this button.
2. User can navigate back to the main movie screen.
3. By clicking this button the user gets navigated to a screen where the image can be viewed in full screen.
4. This label displays relevant information, which was entered in the previous screen. Same follows for the rest of the labels.
5. The user can navigate back to the individual movie screen.

# Rental Screen (Tab 3)



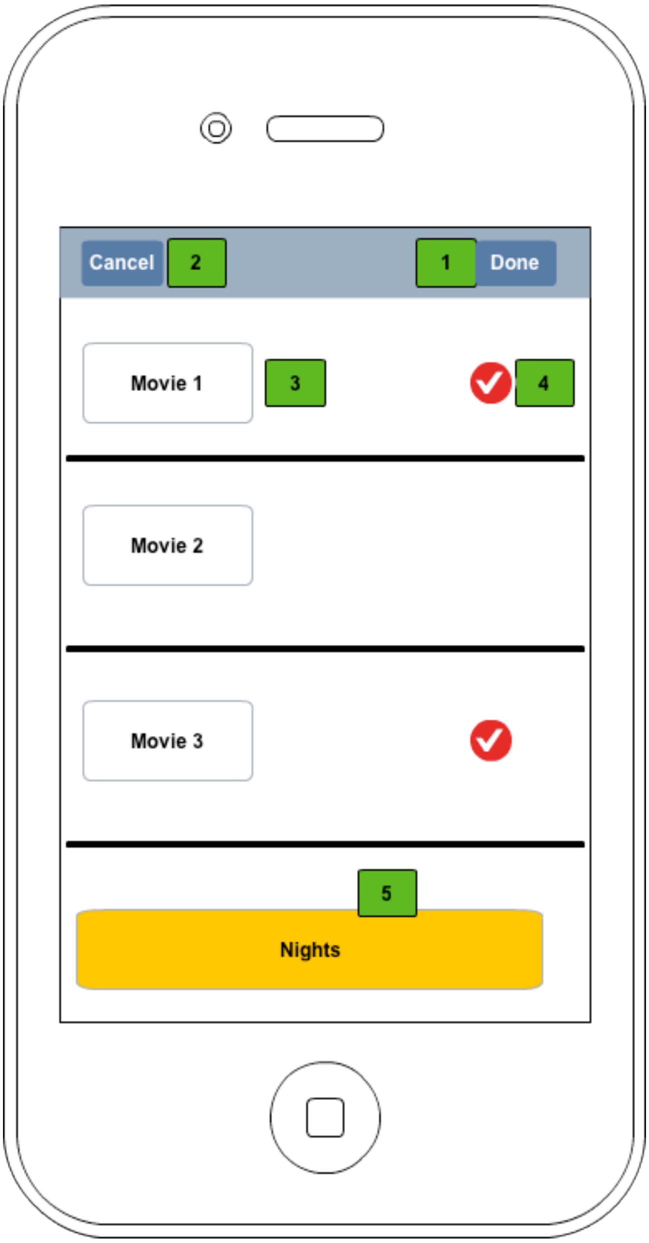
1. All registered customers displayed on this screen. Pressing on the customer name navigates the user to the Current Rental screen, which displays all the movies rented out the currently by the customer.
2. Tab bar to switch between different screens

# Current Rental Screen



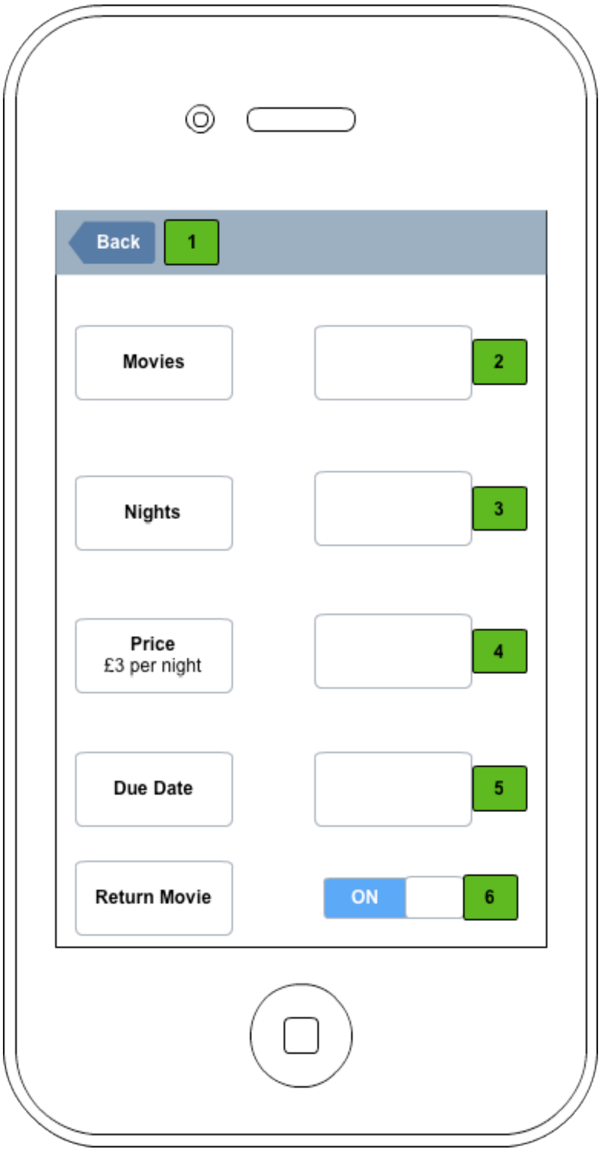
1. Pressing this button navigates the user to the Multiple Movie Select Screen.
2. This will navigate the user back to the Rental Screen.
3. Clicking this area the user gets navigated to the Individual Movie Screen where they also have an option to return the movie.

# Multiple Movie Select



1. Pressing this button saves the selection. While saving a validation is carried out which checks if the customer has rented previously before and if that count is less than 3. If not an alert view appears.
2. Pressing this will cancel the view modally discarding any data input.
3. This label displays the movie name.
4. When multiple movies are selected they are represented with a tick next to the label.
5. A picker view will allow the no. of nights a movie can be rented out.

# Individual Rental Screen



1. This will navigate the user back to the rental screen.
2. This label displays the chosen move.
3. This label displays the number of nights the movie has been rented out for.
4. This label displays the total rental price, which is 3xnumber of nights.
5. This label displays when the movie is due.
6. This switch is used to return the movie. Once the movie is returned that copyID becomes available to be rented out again later on.

# Class Description

## Customer Model Class

1. The account numbers are auto generated using UUID (universally unique identifiers).

2. A function will make use of the relationship between the rental entity and the customer entity. It will check the account state weather any rentals are rented out on this account.

3. There will be a method in this class that deletes the entity if no outstanding rentals are on the account.

4. The movies rented function calculates the total number of movies rented out by a customer. This function will be used for validation when closing the customer account.

## Customer Screen

This controller will display all the registered customers on the screen as a table view.

In the custom table cell the state of the account is displayed using an image. If the account is open a tick appears and vice versa.

## Customer Add Controller

This controller lets you add the movie information in the required text fields.

A validation is carried to see if the user has entered in a customer name, address and postcode. Otherwise an alert view is displayed stopping the user from carrying on. This is validating the text fields in the view therefore the validation is carried out in the view controller.

## Customer Individual Controller

This controller displays the customer information from the model. An extra label is displayed in this controller, which displays the account no. The account numbers are auto generated in the model class. Therefore an option is not available to set an account number when adding in a new customer.

## Customer Edit Controller

In this controller the customer can edit the information previously added in the model via customer add screen.

If the user tries to close the account state a validation is carried out in the model (look in the model class description for more detail)

# Movie Model Class

1. There will be a function that will calculate the total number of copies for any given movie. This will make use of the relationship between movies and copy table. E.g. [self.copies count];

2. Display the copy that are currently rented out. In order to do this I will make use of the relationships between the Rental, Movies and Copy table.

3. When a new movie is added the copy numbers are generated using UUID's.

4. The delete entity function will allow the entity to be deleted if no outstanding rentals are on the account.

# Movie Screen Controller

This controller contains a custom table cell, which displays the movies that are on the system and the stock level of those movies. The stock level calculations are carried out in the movie model class.

# Add Movie Screen

A function will allow a picture to be chosen from the built in photo library to be used as an album art.

If a movie name and quantity have not been entered, an alert view appears requesting the user to fill in the required fields.

While adding the movie a quantity has to be set, this creates the movie copy numbers. These are created using UUID's in the movie model class.

# Individual Movie Screen

This controller displays all the information that is saved in the model.

Pressing the album art button navigates the user to another controller, which displays the album art in full screen.

# Rental Model Class

1. This function calculates the rental due date, which is no of nights + rented date.

2. The return movie function will change the rental state.

# Rental Screen

This controller displays the customers on the system. They can then chose to rent/return movies.

# Current Rental Screen

This screen displays the current movies rented out by the chosen customer. By clicking on each movie label the user is navigated to the individual movie screen, which allows the movie to be returned.

# Multiple Movie Select Screen

On this screen the user is presented with a table view, which displays all the movies that are in stock available to rent. Multiple movies can be selected and at the bottom of the screen a button is displayed which allows the user to select the number of nights the movies can be rented out for, therefore multiple movies can be rented out at once resulting in a pleasant user experience.

# Movie Rental Task

The app must have the following functionality:

Each one of these bullet points will be answered at the end of the document explaining how the task was achieved.

- **Customers can be added**
New customer can be added using the Customer Add Screen. The details entered in the text field are saved in the model. If the user does not enter name, address and postcode a pop up appears preventing the user from continuing.

- **Customers can cancel their account if they have returned all movies**
Edit Customer Screen allows the user to close their account if all the movies have been returned. This makes use of the model, as validation will be carried out on the relationship between the customer table and the rental table.

- **Customers can reopen their account**
In the Edit Customer Screen the user can change the account state back to open via switch. Once the account is re-open the customer can continue to rent movies.

- **A customer can rent and return movies**
The Current Rental Screen allows the customer to navigate between the current rentals and add more movies if desired.
The multiple movie select screen allows the user to pick out multiple movies, which will be added to the current rental list. An option to selection the number of nights is also provided. Once the selection is saved into the model the user can chose to view information on individual movies and also return them in the Individual Rental Screen.

- **A customer has a rental limit of 3 movies**
In the multiple movie select screen the customer can select up to 3 movies. This selection is also checked against what the customer has already rented out. If the number of rentals exceeds the allowed amount an alert view is displayed which prevents the user from continuing.
The relationship which connects the Customer and Rental entity allows to check for this validation. E.g. if customer.rental > 3 display alert view.

- **Each movie has a stock count**
The Copy entity keeps tracks of movie copy numbers. These are auto generated via universally unique identifiers. This is done in the Add Movie Screen where the user can chose the quantity of the new movie being added.

- **Movies can be added to the system**
The movies can be added via Add Movie Screen. All the information entered in the text fields is saved in the model, which can then be displayed in the Individual Movie Screen. Also when adding the movies a quantity needs to be set which will generate the copy id's.

- **Movies can be removed if all copies have been returned**

Edit Movie Screen lets the user delete the movie if all copies of the movie have been returned. This is done via delete button displayed on the Edit Movie screen. This will make use of the relationship between the movies, rental and copy table.

- **Movies have cover art**

Album Art can be added when a new movie is added to the system. In order to add a new album art the user is directed to the build in photo library for the phone, where a choice is given from a selection of pictures in the library. This can later be edited in the Edit Movie Screen. In the Individual Movie Screen the user can also navigate to a view where the album art can be displayed in full screen.