|  |
| --- |
| Project Title  DVD Rental Management System |
| Version Number: 4.0 **Change History**   |  |  |  | | --- | --- | --- | | **Date** | **Author** | **Comments** | | **16/03/13** | **Paul McKenna** | **Summary, assumptions and dependencies added** | | **23/03/13** | **Paul McKenna** | **Added Designs and Diagrams of our project** | | **10/04/13** | **Paul McKenna** | **High Level System Architecture added** | | **15/04/13** | **Paul McKenna** | **Development Methods added** | |
| Summary Design a system that has two levels, Administrators: administrators are allowed perform all system operations and Ordinary users: are only allowed perform ordinary user tasks. Administrators are required to be able to perform the following tasks, Entering the details of a new DVD, Editing the details of any DVD, Deleting a DVD, Adding the details of a customer, Editing the details of any customer, Deleting a customer, Viewing a list of all of the DVD’s that are currently in stock, Viewing a list of all of the DVD’s that are currently overdue. Ordinary users are required to be able to Search for a DVD given a portion of the title, Searching for a user given a portion of their name, Allowing a customer rent a DVD, Allowing a customer return a DVD |

Assumptions & Dependencies

**Language used to develop: Java**

* Extensive knowledge in the language and GUI development in swing.
* Portability over any Operating Systems.
* Graphics libraries: Swing and AWT.
* Disadvantages: Leaked in security in recent release

**Alternatives: Visual Basic.net**

**Advantages:**

* Easy to use, drop and drag development.
* Libraries maintained and updated by Microsoft.

**Disadvantages:**

* Not open source.
* Visual Studio not free, development Licence needed.
* Limits to Windows based operating system.

**Alternatives: C#**

**Advantages:**

* Similar to Java

**Disadvantages:**

* Not Free
* Need developers licence
* Not as many libraries for graphics
* Windows only operating system.

**Software Tools:**

* *Eclipse* – to develop GUI in (development of Android app if pursued).
* *Erwin* - Used to make the Entity Relationship Diagram
* *Star UML* – used to make the Use case, Sequence and Class Diagrams
* *Paint.net* - to develop prototype for screens of website.
* *JUnit* - testing to provide a testing framework.

**Database software tools:**

Options:

* Oracle – Might be able to use
  + Expensive to acquire licence and for maintenance.
  + Secure.
  + Well used and known.
  + Allows for auditing and use of PL/SQL.
* MySQL – Open source
  + Good for android apps and web development.
  + Not as secure.
  + Does not allow for the use of PL/SQL (no auditing).

**End User Characteristics:**

* Easy to use, Intuitive.
* No Technical language
* Colour coded
* Audio impaired: Speaker to call out text selected.
* Visual impaired: Font selection and changes.
* Allow for minimal amount of training.
* Not confusing in any way for any user.

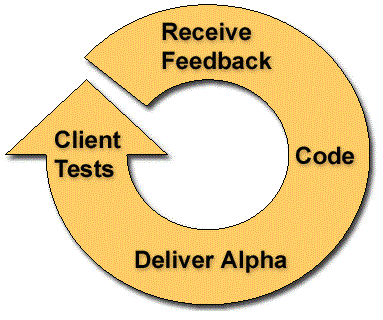
**Possible Changes in Functionality:**

* Addition of a Web Application / website to allow customer.
* Addition of a Smart phone application to allow check and reservation order by customer.
* Reservations implemented to allow members to reserve their DVD before they pick it up in the store. Reservation code will be given to identify reservation and film to be rented.

**Testing Strategy:**

* **Blackbox Testing**
  + Used for testing the program as a unit
  + Inputs put into system and tests for correct outputs
  + Automated tests made to check for errors
* **Whitebox testing**
  + Used to test internal structures of program
  + Will make automated tests also
  + Used to test the database functionality of the system.
* **User Testing**
  + Used to check user reaction to Graphical User Interface
  + Keystroke model
  + Survey on design
  + Test on a range of people, including non-technical, elderly, etc.
* Will allow for incremental testing along the development process.

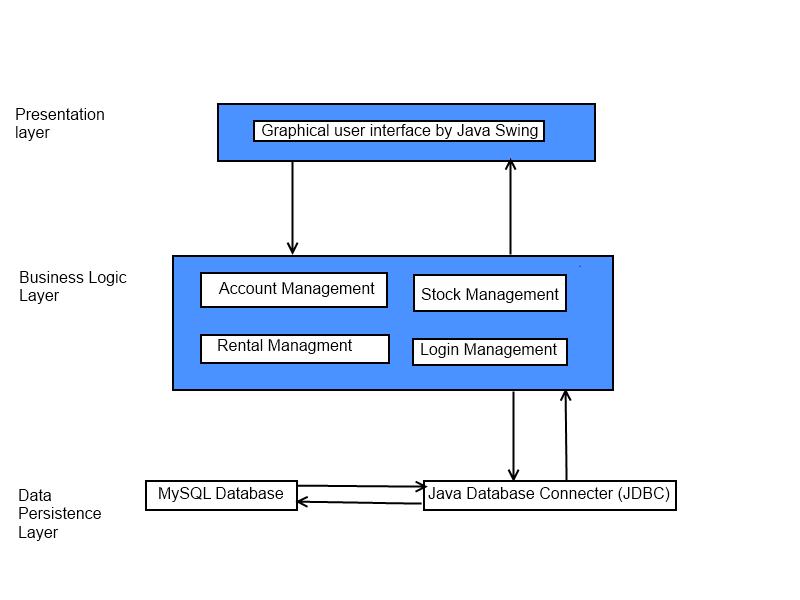
Development Methods



We used an agile development method. We tested any changes we have made at the end of each development. We received feedback to start off. In the first instance it was by what was required in the brief. We then started to code what we feel was wanted. Then using user testing we tested our product and took any feedback from that, the good, the bad and the unneeded. When the feedback has been received we went back into coding to make any changes requested or to improve the thing found good. Again after finishing the coding we “delivered Alpha” which for our project meant we user tested again looking for the same outcome for improvement. After numerous times of receiving feedback, coding, delivering alpha and user testing we improved our product as well as tested in throughout the whole development phrase.

When we first started coding we decided the best thing we could be would be to test as we code. We felt that in the long run it would save us time as when it came to the testing phrase. As well as the fact any changes to the brief that might happen can be changes without disturbing the whole process. The Diagram above shows a rough idea of what we did. Code, Feedback and Code. Another reason why we felt Agile Development was the best choice is because it would give us a good idea of how far along in the project we are.

High Level System Architecture



## Login Screen

The login screen is basic; it has to text fields and a login button. See Figure 1, what login screen does is doing a check on who is logging in. It checks they are a valid user and if they are check what permissions that user has.

## Menu Screen

The Menu Screen is the centre of the system. It links all other features together. It itself doesn’t connect to the database because it doesn’t require and queries. The menu screen consists of five buttons linked to other pages. It also has a menu bar which contains the user manual See Figure 2 and Figure 9

## Settings Screen

Settings are used by the admin only and it used to add a new user to the system. The settings screen only has two buttons add user which brings you into set user screen while the second button returns you to menu screen See Figure 3

## Set New User

The set new user screen had to text fields for the new users name and phone number and a check box for whether the new user is a manager or not. The save Button confirms the changes and inserts the data to the database. The adding the new user is one of the extra features we have decided to add. See Figure 11

## Account Details

This is the screen that is used to search of a customer. You search by the customers name and the information will appear from the database in the text fields. The text fields are uneditable unless a user chooses to edit the details then the text fields will become editable and a save button will appear. When the save button is clicked it updates the database with the new information. From this screen you can also enter a new customer. A pop up window will appear and text fields to enter data. The confirm button enters the data to the database. The Rental History button will open up the rental history of the customer. See Figure 4 and figure 5

## Rental History

This screen will display all DVD that have been rented by that customer. From this screen a user can return a DVD for a customer and also rent a DVD for a customer. If the user chooses either rent or return the text field will become active and the user will enter the name of the DVD to either rent or return and update the list. See Figure 11

## Stock Search

This screen is used to search for stock using its name. Any Stock that has a similar title will appear in the list view. Using this screen you can access Film Details or add a new

DVD title to stock or return to the main menu. See Figure 6

## Stock Details

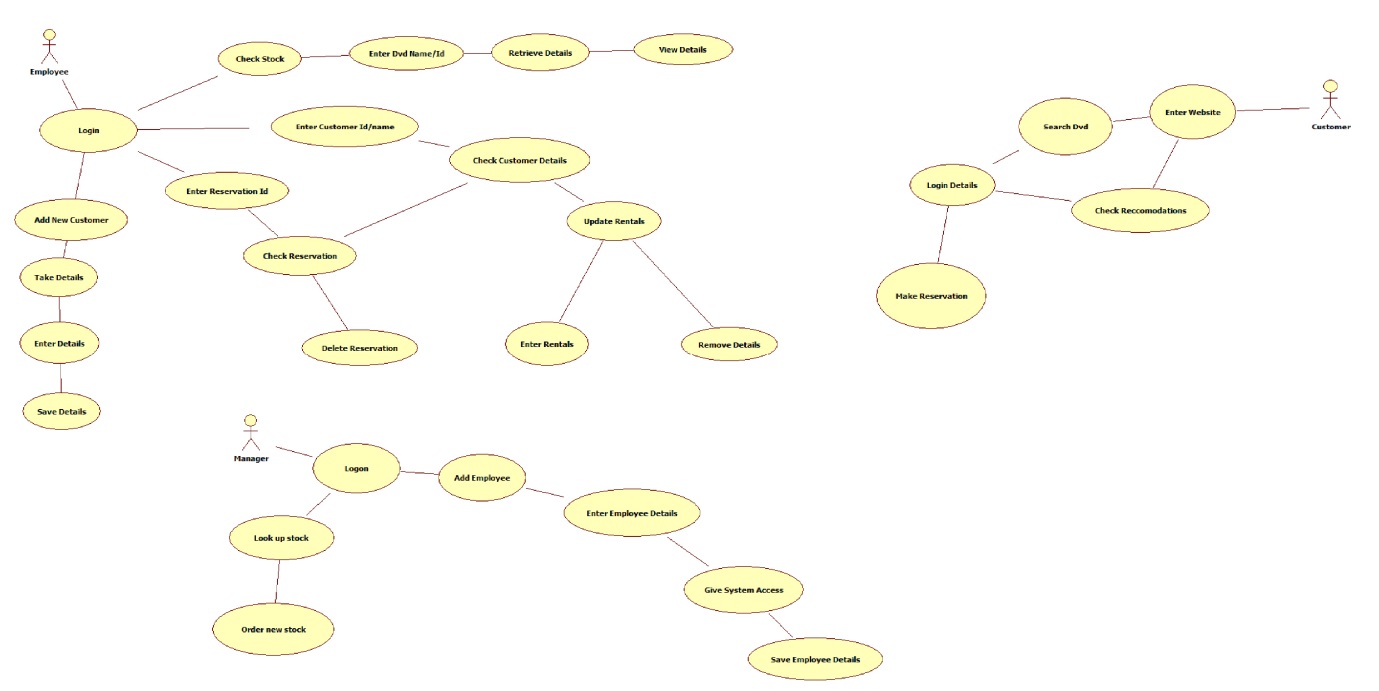
This screen is to display all the details from the stock. Again you search for the film by the title which will display the Director, stock ID, quantity of the stock and the genre. Through this screen you can add to the quantity level, add a new DVD or delete a DVD. See Figure 6

## Reservations

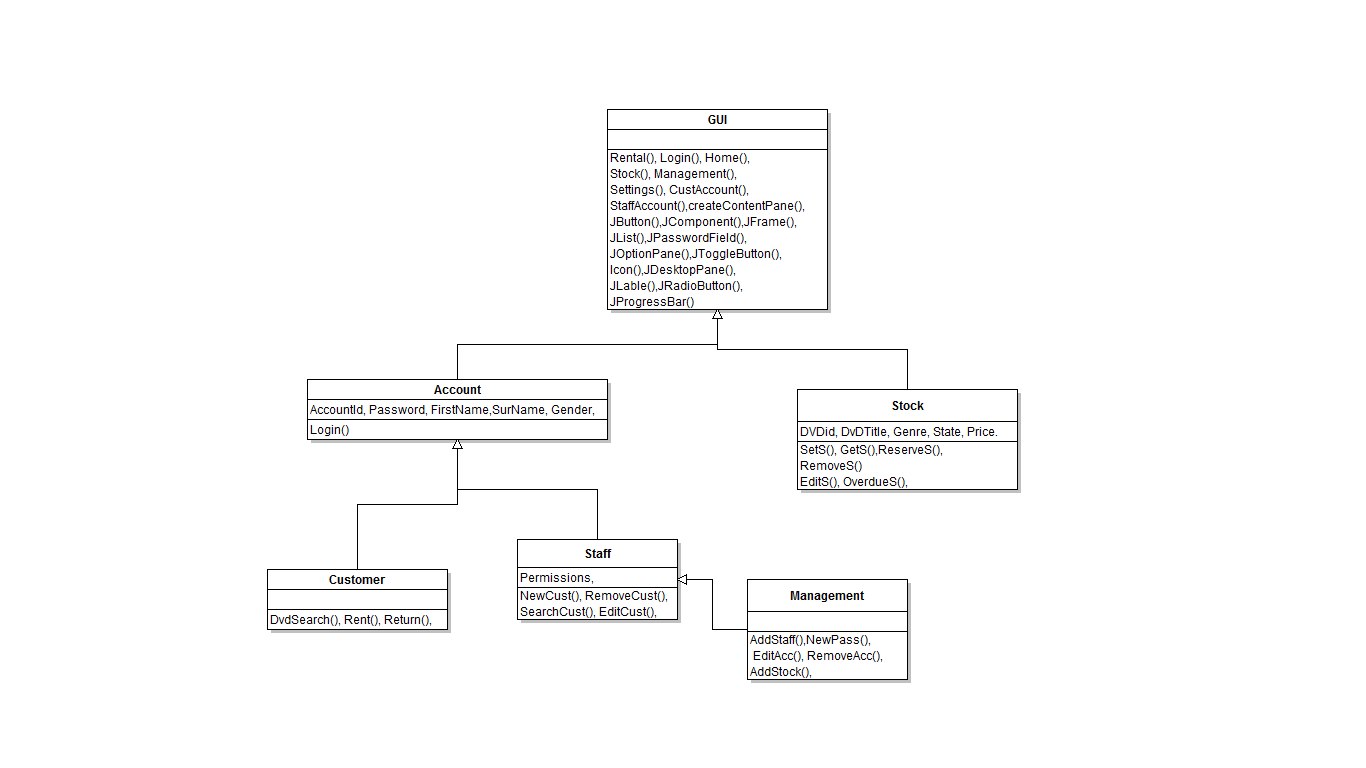
This screen is used to make reservation the user enters the name and number of the customer that wants to make the reservation and clicks confirm. When the reservation is confirmed the two text fields will hold the date the reservation is valid until and the reservation number. See Figure 8

Detailed System Design

## Use case diagram

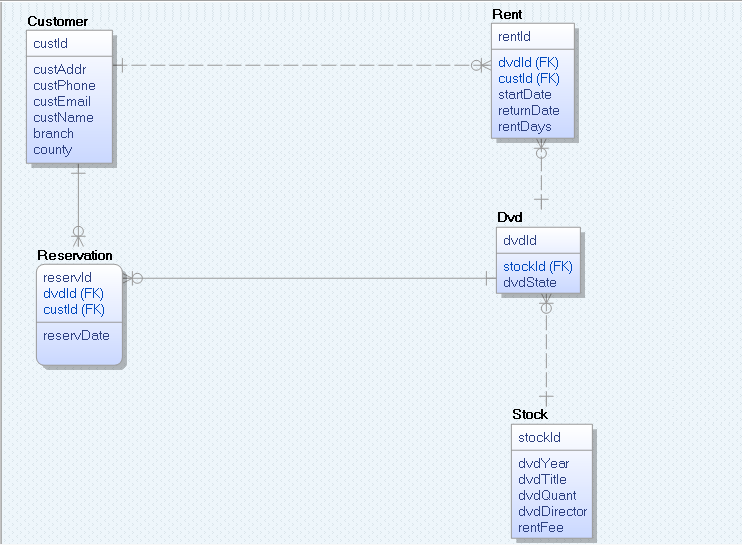


## Class diagram

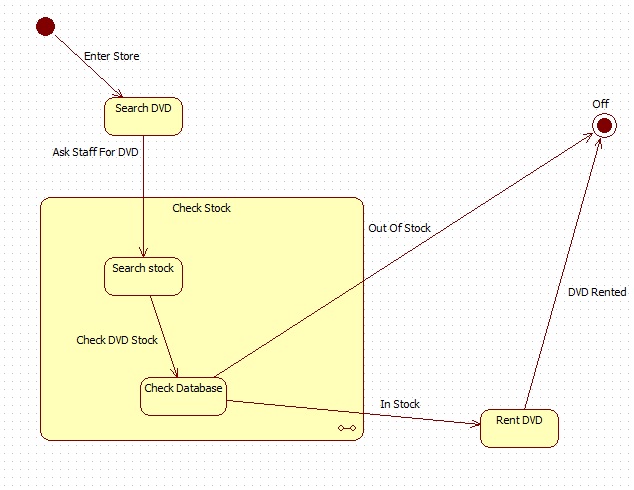


## Sequence diagram

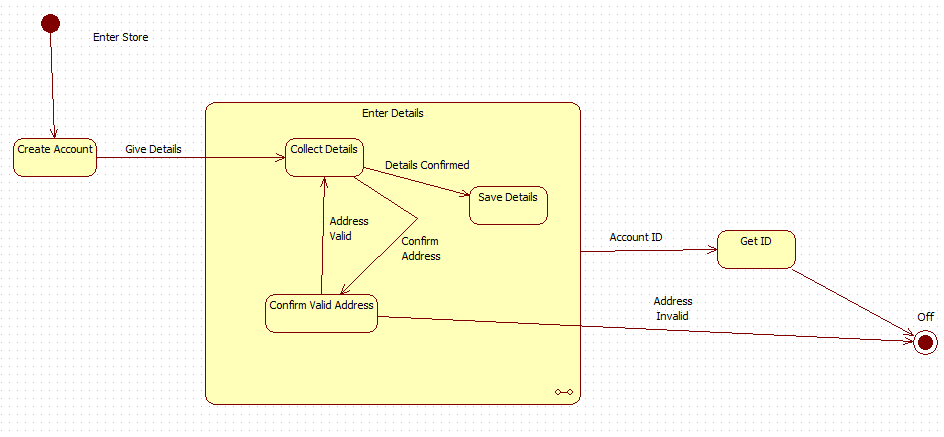
## ERD



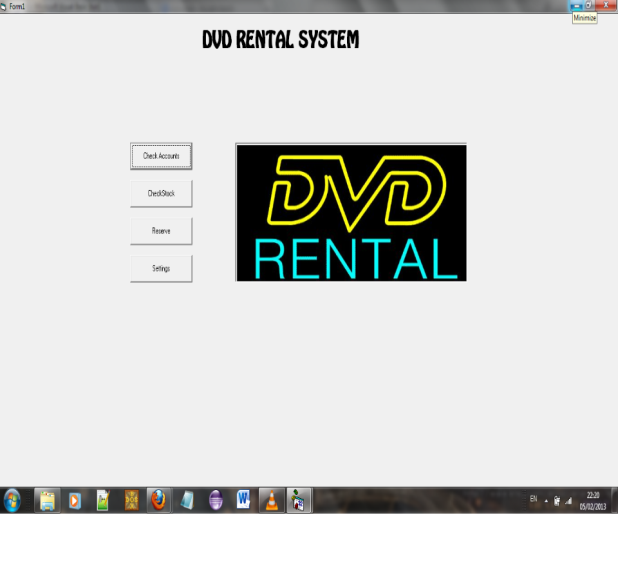
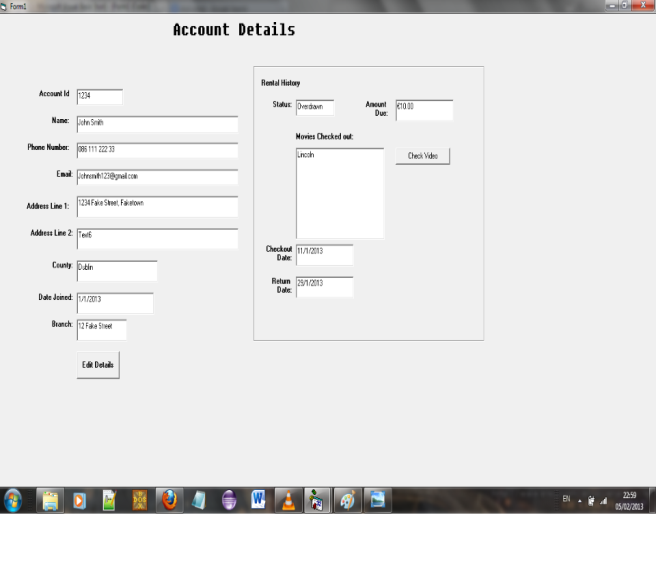
## State transition diagram 1



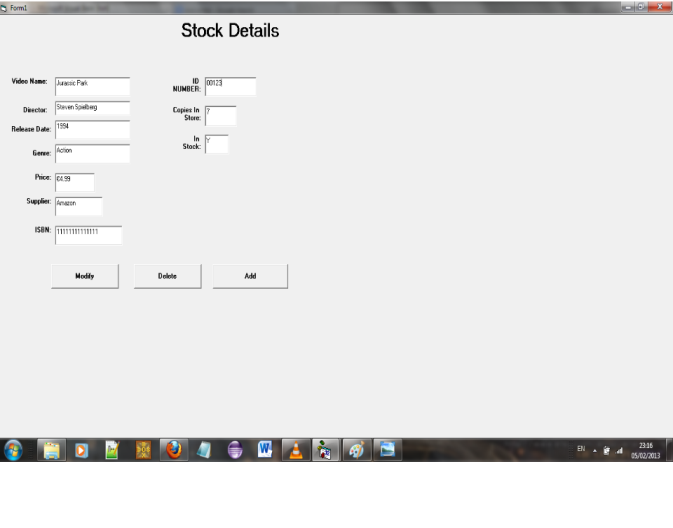
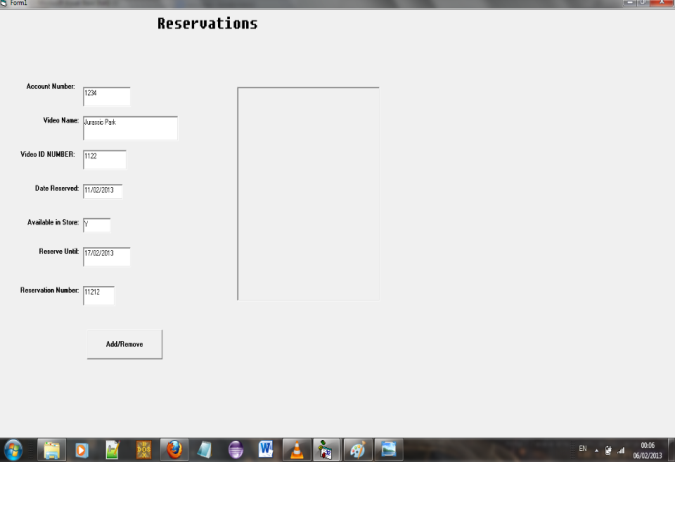
## State transition diagram 2



## Design

Design 1: Login Screen Design 2: Account Details Screen

Design 1: Stock Details Screen Design 1: Reservation Screen

## Screen Shots

Figure 1: The Login Screen Figure 2: The Menu Screen

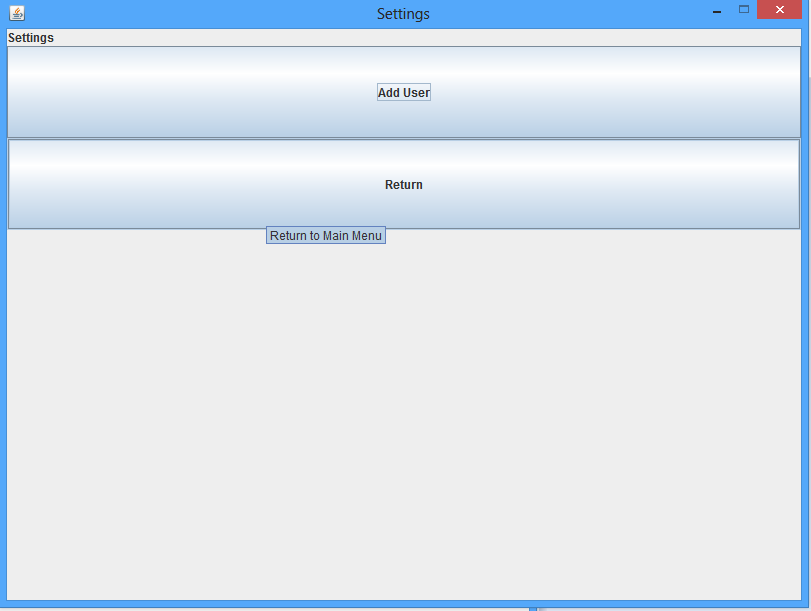
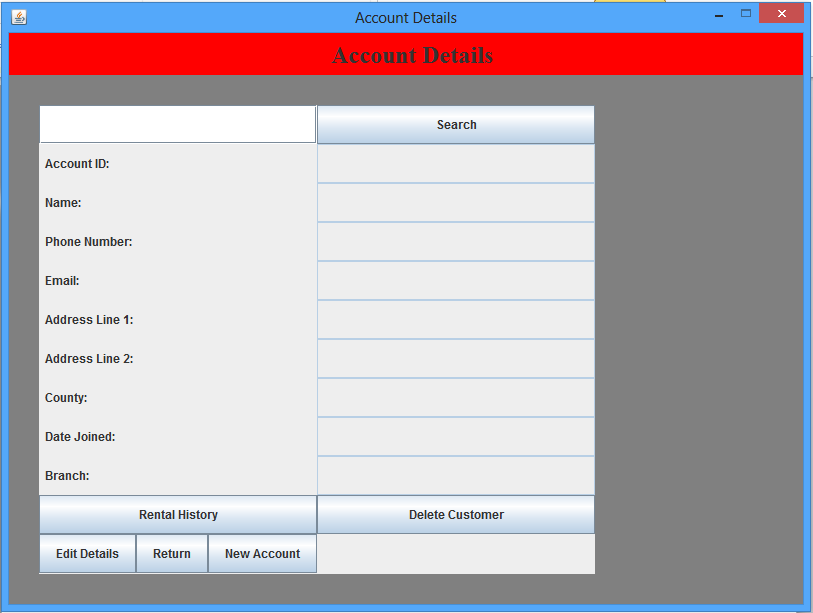
 

Figure 3: Setting screen Figure 4: Account Details Screen

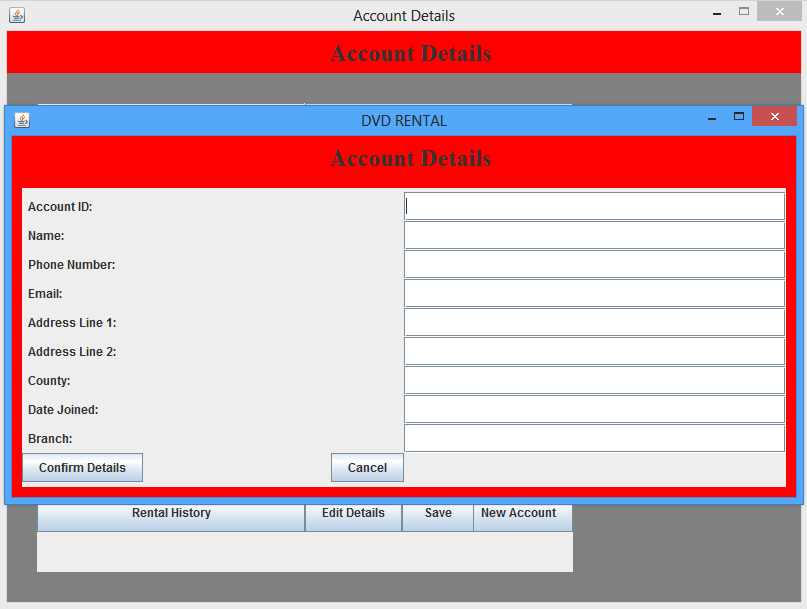
 

Figure 5: New Account Screen Figure 6: Stock Details Screen

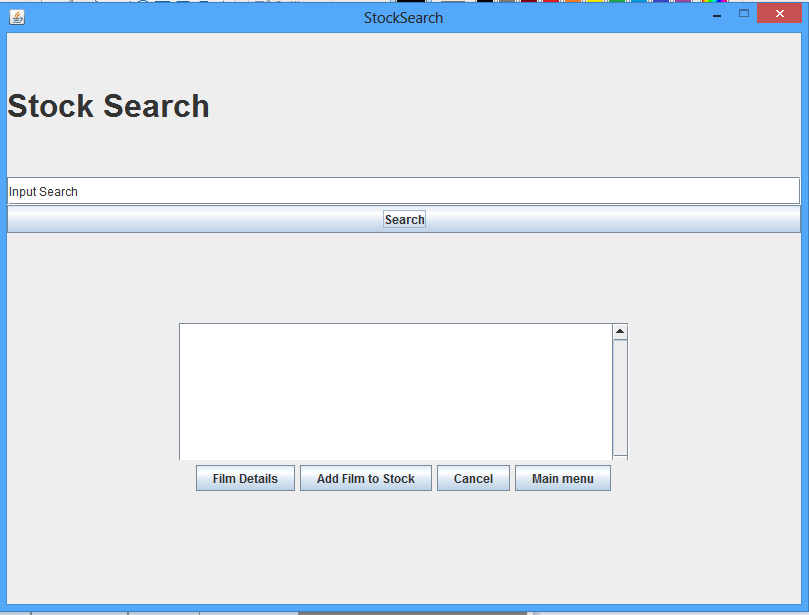
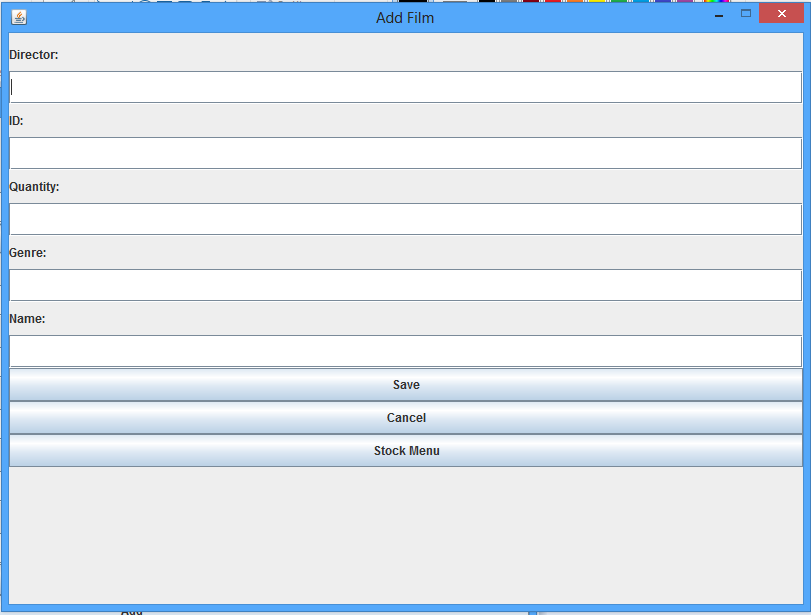
 

Figure 6: Stock Search Screen Figure 7: Add Film Screen

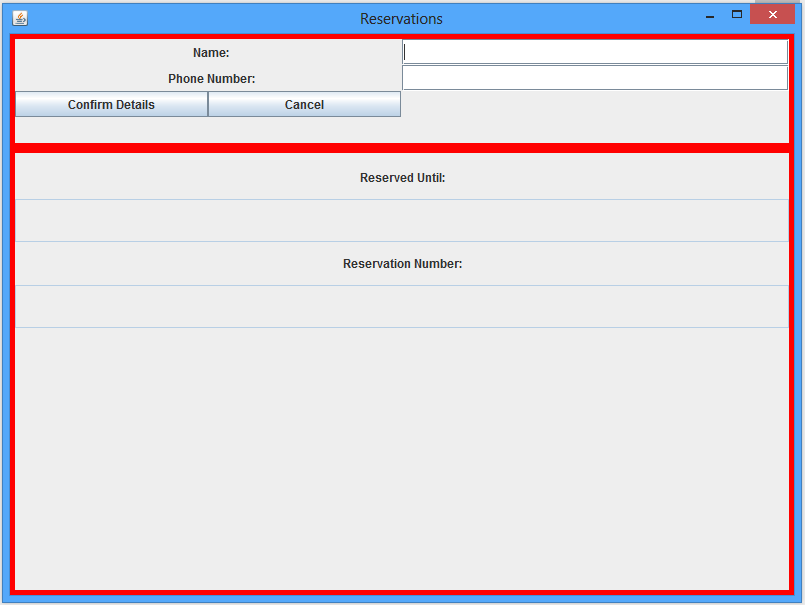
 

Figure 8: Reservation Screen Figure 9: About Screen

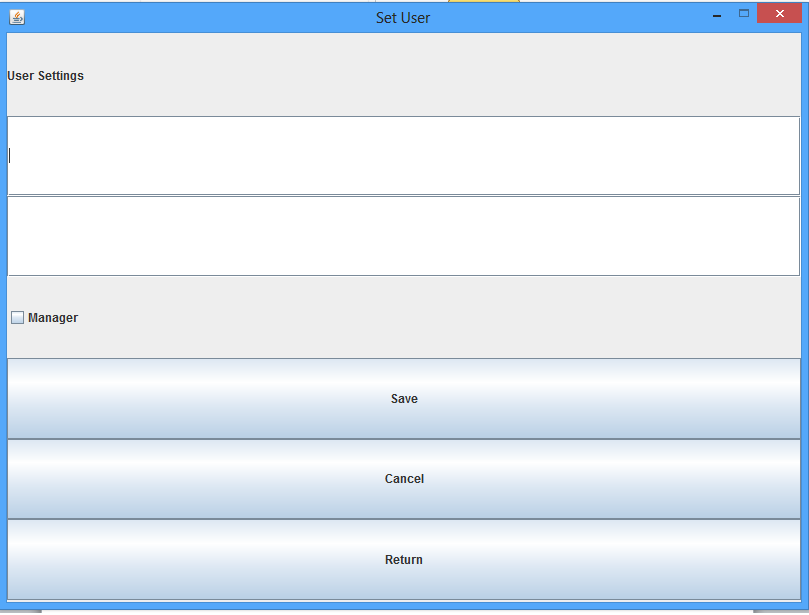
 

Figure 10: Logout Screen Figure 11: Set New User Screen

## 

Figure 12: Rental History

## Related Documents

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Document Title** | **Author(s)** | **Description** | | Design Document | Paul | This is the Document that holds all the designs or any changes that have been made | | Test Plan | Dennis | This is the test plan we are using. It holds what plans we are using as well as test strategies. | | Minutes/Agenda | All | The minutes and must important points of discussed in the meetings | | User Manual | Dennis | This is a description on how to use our system | |