Requirements Engineering

TvLicenceSYS

Submitted By: Ibrahim Khalid(t00218562)

Computing with Software Development

Date Submitted: 18/12/2021

**Table of Contents**

[1. Introduction/overview 3](#_Toc101569697)

[2. Functional Components 4](#_Toc101569698)

[3. User Requirements 5](#_Toc101569699)

[3.1. TvLicenceSYS will manage Purchases 5](#_Toc101569700)

[3.2. TvLicenceSYS will process Renewals 5](#_Toc101569701)

[3.3. TvLicenceSYS will provide Admin functionality 5](#_Toc101569702)

[4. System Requirements 6](#_Toc101569703)

[4.1. System Level Use Case Diagram 6](#_Toc101569704)

[4.2. Manage Purchases 7](#_Toc101569705)

[4.2.1. Purchase Licence 7](#_Toc101569706)

[4.2.2. Transfer Licence 11](#_Toc101569707)

[4.2.3. Cancel Licence 14](#_Toc101569708)

[4.3. Manage Renewals 16](#_Toc101569709)

[4.3.1. Issue renewal notice 16](#_Toc101569710)

[4.3.2. Renew licence 18](#_Toc101569711)

[4.3.3. List overdue licences 22](#_Toc101569712)

[4.3.4. Produce property history 24](#_Toc101569713)

[4.4. Admin 26](#_Toc101569714)

[4.4.1. Yearly revenue analysis 26](#_Toc101569715)

[4.4.2. Yearly authority analysis 28](#_Toc101569716)

[5. System Model 30](#_Toc101569717)

[5.1. Level-0 DFD 30](#_Toc101569718)

[5.2. Level-1 DFD 31](#_Toc101569719)

[5.3. Level-2 DFD (Process P1: Purchase) 32](#_Toc101569720)

[5.4. Level-2 DFD (Process P2: Renewal) 33](#_Toc101569721)

[5.5. Level-2 DFD (Process P3: Admin) 34](#_Toc101569722)

[6. Data Model (Class Diagram) 35](#_Toc101569723)

[6.1. Class Diagram 35](#_Toc101569724)

[6.2. Relational Schema 35](#_Toc101569725)

[6.3. Database Schema 36](#_Toc101569726)

[7. Conclusion 38](#_Toc101569727)

[8. Appendices 38](#_Toc101569728)

[8.1. Appendix A – Example screenshots 38](#_Toc101569729)

# Introduction/overview

The system manages the tv licenses of Ireland.

The system allows users to add, cancel or update a licence.

The system allows the admin to issue renewal notices from a list of expired or soon to be expired licenses, admin will be able to view the licence history of an address.

The system allows the admin to view revenue analysis for a selected year in a table format with details of each month.

The system will be connected to a database which allows the system to store the data generated.

Admin login is “admin123”.

# Functional Components

# User Requirements

## TvLicenceSYS will manage Purchases

* + 1. TvLicenceSYS will add a licence.
    2. TvLicenceSYS will update the transfer of a licence.
    3. TvLicenceSYS will update the cancelling of a licence.

## TvLicenceSYS will process Renewals

* + 1. TvLicenceSYS will produce soon to be expired licenses.
    2. TvLicenceSYS will update a renewed licence.
    3. TvLicenceSYS will produce overdue licenses.
    4. TvLicenceSYS will produce property history of purchased licence.

## TvLicenceSYS will provide Admin functionality

* + 1. TvLicenceSYS will produce a yearly revenue analysis.
    2. TvLicenceSYS will produce a yearly authority analysis.

# System Requirements

## System Level Use Case Diagram

The following system level use case diagram illustrates the high-level system requirements.

Admin

User

## Manage Purchases

This module provides functions to

### Purchase Licence

This function records the details of a licence which is purchased by a customer.

User

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Purchase Licence** | |
| **Use Case Id** | TV001 | |
| **Priority** | 1 | |
| **Source** | System Purchases | |
| **Primary Business Actor** | User | |
| **Other Participating Actors** |  | |
| **Description** | This function will add a new licence and a new customer or update a previous customer to the database | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **User** | **System** |
|  | **Step 1:** Customer invokes the Purchase Licence function.  **Step 3:** Customer enters the required data:   * First name * Last name * Address street * Town * County * Email * Card number * Card name * Expiry date * Cvv   **Step 4:** user clicks apply button  **Step 7:** User clicks Yes | **Step 2:** the system displays UI  **Step 5:** The system validates:   * All fields must not be empty * First name is and last name is all alphabetic * Street is alphanumeric * Town length is greater then three and is all alphabetic * Email is of valid standard * Card number length is 16 and is all digits * Card name is a full name * Expiry date is a valid date in the future * Cvv length is three * The entered street, town and county those not have a current valid licence   **Step 6**: System will ask with confirmation box with the price of a licence, if the user wants to continue.  **Step 8:** Add data to customer:   * CusId * FirstName * LastName * Email   **Step 9:** add data to licence file:   * LicenceId * Cus\_Id * Street * Town * County * ExpiryDate * Fee * Stat   **Step 10:** Reset UI  **Step 12:** Show user their licence id and customer id. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid data detected**  **Address has licence**  **User clicks ‘no’ in the confirmation box** | **Step 4:** Customer enters invalid data:   * First name * Last name * Address street * Town * County * Email * Card number * Card name * Expiry date * Cvv   **Step 5:** user clicks apply button  **Step 8:** user clicks no | **Step 6:** System validates  **Step 7:** systemdisplays error message and where the error happened  **Step 5:** system validates  **Step 6:** System displays message that licence already exist for this account and send user their licence id and customer id to the email associated to that licence  **Step 9:** system clears inputted data |
| **Conclusions** | Licence and customer has been added to the database | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Transfer Licence

This function transfers User Tv licence to a new address of that user.

User

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Transfer Licence** | |
| **Use Case Id** | TV002 | |
| **Priority** | 1 | |
| **Source** | System Purchases | |
| **Primary Business Actor** | User | |
| **Other Participating Actors** |  | |
| **Description** | This function will update the current street, town and county of the user licence. | |
| **Preconditions** | The user must have a valid in date licence. | |
| **Trigger** |  | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1:** User invokes the Transfer Licence function.  **Step 3:** User enters required data:   * Licence number   **Step 8:** User clicks confirm  **Step 9:** User enters required data:   * Address street * Town * county | **Step 2:** the system displays UI  **Step 4:** System validates:   * Licence number is digits   **Step 5:** System checks licence file if licence id exists.  **Step 6:** system will display user info from database and confirmation button:   * Full name * Street * Town * County * Expiry date   **Step 7:** System displays new address text areas.  **Step 10**: System validates:   * Street is alphanumeric * Town length is greater then three and is alphabetic * A county has been selected from the combo box   **Step 11:** Update data in licence file:   * Street * Town * County   **Step 12:** System shows message to user that their licence has been transferred.  **Step 13:** System resets UI |
| **Alternate Scenarios** | **User** | **System Response** |
| **Invalid licence id entered**  **Licence does not exist**  **Invalid address data entered** | **Step 3:** User enters invalid data:   * Licence number   **Step 3:** User enters required data:   * Licence number   **Step 10:** User enters invalid data:   * Address street * Town * county | **Step 4:** System validates:  **Step 5:** System shows message to user that licence number must be digits.    **Step 4:** System validates  **Step 5:** Licence id does not exist  **Step 6:** System will show message to user that licence id was not found  **Step 11:** System validates  **Step 12:** System shows message that data entered is invalid and where the error happened. |
| **Conclusions** | Licence will be updated with new address. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Cancel Licence

This function is for when the user wants to cancel their tv licence.

User

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Cancel Licence** | |
| **Use Case Id** | TV003 | |
| **Priority** | 1 | |
| **Source** | System Purchases | |
| **Primary Business Actor** | User | |
| **Other Participating Actors** |  | |
| **Description** | This function will update a licence stat to cancelled | |
| **Preconditions** | The user must have a valid licence. | |
| **Trigger** |  | |
| **Expected Scenario** | **User** | **System** |
|  | **Step 1:** User invokes the cancel licence function.  **Step 3:** User enters the required data:   * Licence number   **Step 7:** User clicks confirm | **Step 2:** the system displays UI  **Step 4:** System validates:   * Licence number is digits   **Step 5:** System checks licence file if licence id exists.  **Step 6:** system will display user info from database and confirmation button:   * Full name * Street * Town * County * Expiry date   **Step 8**: System checks if licence stat is valid.  **Step 9:** System updates licence   * Stat   **Step 10:** System resets UI  **Step 11:** System shows user message that licence has been successfully cancelled |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid licence id entered**  **Licence does not exist**  **Licence stat is invalid** | **Step 3:** User enters invalid data:   * Licence number   **Step 3:** User enters required data:   * Licence number | **Step 4:** System validates:  **Step 5:** System shows message to user that licence number must be digits.  **Step 4:** System validates  **Step 5:** Licence id does not exist  **Step 6:** System will show message to user that licence id  **Step 8:** System checks if licence stat is valid.  **Step 9:** System shows message to user that licence is already cancelled or expired |
| **Conclusions** | If valid, User licences will be cancelled. The address associated will no longer have a valid licence in the database | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Manage Renewals

This module will provide functions to.

### Issue renewal notice

This function will issue a notice message to user that their licence needs renewal.

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Issue renewal notice** | |
| **Use Case Id** | TV004 | |
| **Priority** | 1 | |
| **Source** | Manage Renewals | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | This function will produce a list of licenses that will expire within the month | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** |  | **System** |
|  | **Step 1:** Admin invokes the overdue licence’s function.  **Step 3:** Admin clicks load function  **Step 5:** Admin clicks send notice button | **Step 2:** the system displays according UI.  **Step 4:** System produces list of licenses with expiry date within the next month  **Step 6:** System updates licence   * Stat   **Step 7:** System emails all the accounts that where on the list  **Step 8:** System shows message to admin that a notice has been sent |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid Data Entered** |  |  |
| **Conclusions** | Admin has seen all overdue licenses. Licence stat will be changes to soon, notice will be sent to accounts | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Renew licence

This function will allow user to renew their current licence

User

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Renewal Licence** | |
| **Use Case Id** | TV005 | |
| **Priority** | 1 | |
| **Source** | Manage Renewals | |
| **Primary Business Actor** | User | |
| **Other Participating Actors** |  | |
| **Description** | This function will update the licence of a current user and add a licence to the history file. | |
| **Preconditions** | The user must have a valid in date licence. | |
| **Trigger** |  | |
| **Expected Scenario** | **User** | **System** |
|  | **Step 1:** User invokes the renew licence function.  **Step 3:** User enters required data:   * User id   **Step 9:** User enters required data:   * Card number * Card name * Expiry date * Cvv   **Step 10:** User clicks on renew button  **Step 13:** User clicks ok | **Step 2:** the system displays according UI.  **Step 4:** System validates:   * customer number is digits   **Step 5:** System checks Customer file if i.d exists.  **Step 6:** System checks if licence stat is not valid  **Step 7:** System display textboxes  **Step 8:** System fills in user details:   * First name * Last Name * Email   **Step 11:** System validate data:   * Email is valid format * First and last name is alphabetic * Card number length is 16 and is all digits * Card name is a full name * Expiry date is a valid date in the future * Cvv length is three   **Step 12**: System will ask with confirmation box with the price of a licence, if the user wants to continue.  **Step 14:** System will add to history file:   * hisId * cusId * street * town * county * expiryDate * fee   **Step 15:** System will update customer file:   * firstName * lastName * Email   **Step 16:** System will update licence file:   * expiry date * fee * stat   **Step 17:** System resets UI  **Step 18:** system will show confirmation message with new expiry date to use |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid licence id entered**  **Licence does not exist**  **Licence stat is invalid**  **Invalid enters invalid data**  **User clicks ‘no’ in the confirmation box** | **Step 3:** User enters invalid data:   * Licence number   **Step 3:** User enters required data:   * Licence number   **Step 9:** User enters required data:   * Card number * Card name * Expiry date * Cvv   **Step 10:** User clicks on renew button  **Step 13**: User clicks no | **Step 4:** System validates:  **Step 5:** System shows message to user that licence number must be digits.    **Step 4:** System validates  **Step 5:** Customer id does not exist  **Step 6:** System will show message to user that licence id was not found  **Step 6:** System checks if licence stat is valid.  **Step 7:** System shows message to user that licence is already valid  **Step 11:** System validates  **Step 12:** System shows message that data entered is invalid and where the error happened.  **Step 4:** system clears inputted data |
| **Conclusions** | Licence will be updated, history will be added and customer will be updated. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### List overdue licences

This function will list overdue licences to the admin

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **List overdue licences** | |
| **Use Case Id** | TV006 | |
| **Priority** | 1 | |
| **Source** | Manage Renewals | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | This function will produce a list of overdue licenses | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **User** | **System** |
|  | **Step 1:** Admin invokes the overdue licence’s function.  **Step 3:** Admin clicks load function  **Step 5:** Admin clicks send notice button | **Step 2:** the system displays according UI.  **Step 4:** System produces list of licenses with expiry date less then todays date  **Step 6:** System updates licence   * Stat   **Step 7:** System emails all the accounts that where on the list  **Step 8:** System shows message to admin that a notice has been sent |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid Data Entered** |  |  |
| **Conclusions** | Admin has seen all overdue licenses. Licence stat will be changes to expired | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Produce property history

This function will allow user and admin to look at the history of licence of a property

User

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Produce property history** | |
| **Use Case Id** | TV007 | |
| **Priority** | 1 | |
| **Source** | Manage Renewals | |
| **Primary Business Actor** | User | |
| **Other Participating Actors** | Admin | |
| **Description** | This function allows user and admin to view tv licence history for address property, | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **User, admin** | **System** |
|  | **Step 1:** User or admin invokes the renew licence function.  **Step 3:** User enters required data:   * Street * Town * County | **Step 2:** the system displays according UI.  **Step 4:** System validates data:   * Street, town and county are not empty   **Step 5:** System produces the following:   * Cus\_Id * expiryDate * startDate * firstName * lastName |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid Data Entered** | **Step 3:** User enters required data:   * Street * Town * County | **Step 4:** System validates data:  **Step 5:** System displays message that user must enter all fields |
| **Conclusions** | User or admin have view of tv licence history of an address. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Admin

This module will allow provide function to.

### Yearly revenue analysis

This function will provide revenue analysis to the admin

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Yearly revenue analysis** | |
| **Use Case Id** | TV008 | |
| **Priority** | 1 | |
| **Source** | Manage Admin | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | This function will generate a yearly revenue report for the admin. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Admin** | **System** |
|  | **Step 1:** Admin invokes the yearly renew function.  **Step 3:** Admin chooses a year from the combo box | **Step 2:** the system displays according UI.  **Step 4:** System produces a list of months and income from that month with the sum on the bottom of the list  **Step 5**: System highlights green the month with most income and red the month with the least income |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid Data Entered** |  | . |
| **Conclusions** | Admin will receive full report on the revenue made for the corresponding year. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Yearly authority analysis

This function will provide authority analysis to the admin.

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Yearly authority analysis** | |
| **Use Case Id** | TV009 | |
| **Priority** | 1 | |
| **Source** | Manage Admin | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | This function allows user and admin to view tv licence history for address property, | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **User, admin** | **System** |
|  | **Step 1:** Admin invokes the yearly renew function.   * **Step 3:** Admin chooses a year from the combo box | **Step 2:** the system displays according UI.  **Step 4:** System produces a list of months and income from that month with the sum on the bottom of the list  **Step 5**: System highlights green the month with most income and red the month with the least income |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid Data Entered** |  |  |
| **Conclusions** | Admin will receive full report on the revenue made for the corresponding year. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

# System Model

The following dataflow diagrams have been produced for the system:

## Level-0 DFD

Diagram

Description automatically generated

## Level-1 DFD

Diagram

Description automatically generated

## Level-2 DFD (Process P1: Purchase)

Diagram

Description automatically generated

## Diagram Description automatically generatedLevel-2 DFD (Process P2: Renewal)

## Level-2 DFD (Process P3: Admin)

Diagram

Description automatically generated

# Data Model (Class Diagram)

There are three classes to be made, the database has a customer file which holds the user’s name and email. The licence file has the address, expiry date of the licence, the cost of the licence and the status of the licence. The history file contains the address, this is not duplication of data as a licence can be transferred to a different address so it will not always be the same as the licence file data, it also has the expiry date of that licence and the cost.

## A screenshot of a computer Description automatically generated with low confidenceClass Diagram

## Relational Schema

Relational schema for the data requirements

Customer(CusID, FirstName, LastName, email)

History(HistoryId, Street, Town, County, ExpiryDate, Fee, Cus\_Id)

History(LicenceId, Street, Town, County, ExpiryDate, Fee, stat, Cus\_Id)

History(StartDate, Fee)

## Database Schema

A definition of the database to be implemented.

This includes primary key, foreign key and other constraints to be implemented.

**Schema**: TvLicenceSYS

**Relation** *Customer*

CusId NUMBER(8) NOT NULL

FirstName VARCHAR(30) NOT NULL

LastName VARCHAR(30) NOT NULL

Email VARCHAR(100) NOT NULL

**Primary Key:** CusId

**Relation** *History*

HistoryId NUMBER(10) NOT NULL

Cus\_Id NUMBER(8) NOT NULL

Street VARCHAR(50) NOT NULL

Town VARCHAR(50) NOT NULL

County VARCHAR(50) NOT NULL

ExpiryDate VARCHAR(20)

Fee numeric(5,2) CHECK (Fee > 0)

**Primary Key:** HistoryId

**[Foreign Key:** Cus\_Id **References** *Customer*]

**Relation** *Licence*

LicenceId NUMBER(10) NOT NULL

Cus\_Id NUMBER(8) NOT NULL

Street VARCHAR(50) NOT NULL

Town VARCHAR(50) NOT NULL

County VARCHAR(50) NOT NULL

ExpiryDate VARCHAR(20)

Fee numeric(5,2) CHECK (Fee > 0)

Stat VARCHAR(9) CHECK (Stat IN ('Valid', 'Cancelled', 'Soon', 'Expired'))

**Primary Key:** LicenceId

**[Foreign Key:** Cus\_Id **References** *Customer*]

**Relation** *CurrentFee*

StartDate Date

Fee numeric(5,2)

**Primary Key:** StartDate

# Conclusion

The system works with full functionality, I would add some more features, allowing the user to view their current licence with all it details and customer details, assigning a 10-character long licence id and a bar chart for yearly authority analysis.

The system can add, update and produce data from the database.

More UI design can be done to the system.

# Appendices

## Appendix A – Example screenshots

















