**Property rental System**

Software Engineering: Catherine Woods

Submitted By: Peter Mikulasko (MA\_Y2)

Submitted on: 20 April 2015

Table of Contents

[**1.** **Introduction** 4](#_Toc431292882)

[**Functional Components** 5](#_Toc431292883)

[**2.** **User Requirements** 6](#_Toc431292884)

[**3.1.** **The system will manage tenants** 6](#_Toc431292885)

[**3.2.** **The system will manage properties** 6](#_Toc431292886)

[**3.3.** **The system will manage staff** 6](#_Toc431292887)

[**3.4.** **The system will process Rentals** 6](#_Toc431292888)

[**3.** **System Requirements** 8](#_Toc431292889)

[**4.** **System Level Use Case Diagram** 8](#_Toc431292890)

[**4.1.** **Manage Tenants** 9](#_Toc431292891)

[4.1.1. Amend Tenant 9](#_Toc431292892)

[4.1.2. Remove Tenant 11](#_Toc431292893)

[4.1.3. List Tenants 13](#_Toc431292894)

[4.1.4. Query tenants 14](#_Toc431292895)

[**4.2.** **Manage Staff** 15](#_Toc431292896)

[4.2.1. Add Staff 15](#_Toc431292897)

[4.3. **Manage Properties** 16](#_Toc431292898)

[4.3.1. Add Property 16](#_Toc431292899)

[4.4. Amend Property 18](#_Toc431292900)

[**4.5.** **Manage Staff** 21](#_Toc431292901)

[4.5.1. Add Staff 21](#_Toc431292902)

[4.5.2. Remove Staff 22](#_Toc431292903)

[4.6. **Manage Rentals** 24](#_Toc431292904)

[4.6.1. Rent Property 24](#_Toc431292905)

[4.6.2. Terminate Rental 25](#_Toc431292906)

[4.6.3. Record Payment 25](#_Toc431292907)

[4.6.4. Issue Late Notice 25](#_Toc431292908)

[4.6.5. Return Deposit 25](#_Toc431292909)

[**5.** **System Model(data flow diagram)** 26](#_Toc431292910)

[**5.1.** **DFD Elements** 26](#_Toc431292911)

[**5.2.** **Level 0 DFD** 27](#_Toc431292912)

[**5.3.** **Level 1 DFD** 28](#_Toc431292913)

[**5.4.** **Level 2 DFD** 29](#_Toc431292914)

[**6.** **Data Model** 31](#_Toc431292915)

[**7.** **Relational Schema** 32](#_Toc431292916)

[**8.** **Database Schema** 33](#_Toc431292917)

[**9.** **Program Specifications** 36](#_Toc431292918)

[**10.** **Conclusion** 37](#_Toc431292919)

[**11.** **References:** 38](#_Toc431292920)

# **Introduction**

Property rental System introduction….

# **Functional Components**

Property Rental System Hierarchy Chart

# **User Requirements**

Intro to User requirements sys

## **The system will manage tenants**

* + 1. The system will be able to register new tenant
    2. The system will allow tenant details to be amended
    3. The system will allow tenant details to be removed

## **The system will manage properties**

* + 1. The system will be able to add new properties
    2. The system will be able to amend existing properties
    3. The system will allow to query all properties

## **The system will process Rentals**

* + 1. The system will be able to Rent property
    2. The system will be able to Terminate rental
    3. The system will be able to Record payment
    4. The system will allow Return deposit

1. **Admin**
2. The system will be able to display Income Analysis
3. The system will be able to display Property Analysis
4. The system will be able to display Rental Analysis

# **System Requirements**

Intro to system requirements sys

# **System Level Use Case Diagram**

Staff

Staff

|  |
| --- |
|  |

## **Manage Tenants**

Manage tenants will allow system to perform multiple tasks on tenants. System will be able add(register) new tenants to the system. Amend tenants, remove tenants from the system

Register tenant

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Register Tenant** | **Author:** |
| **Use Case Id** | 1 | **Date:** |
| **Priority** | 1 | |
| **Source** | Staff | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | This function register new tenant to the system | |
| **Preconditions** |  | |
| **Trigger** | Tenants is registered to the system | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**Staff wants to register new tenant to the system  **Step 3:**Staff query tenant  **Step4:** Staff edit tenants details | **Step 2:**System displays UI  **Step 4:**The system process query  **Step5:** System validates data  **Step6:** System updates tenants details in tenants file |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid Data Entered** |  | Step 6:The system displays an appropriate error message. |
|  |  | |
| **Conclusions** | Tenants details are changed | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

1. Amend Tenant

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Amend Tenant** | **Author:** |
| **Use Case Id** | 1 | **Date:** |
| **Priority** | 1 | |
| **Source** | Staff | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | This function amends the details of a tenant | |
| **Preconditions** | Tenant must be registered in the system | |
| **Trigger** | Tenants information amended | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**Staff wants to amend tenants information  **Step 3:**Staff query tenant  **Step4:** Staff edit tenants details | **Step 2:**System displays UI  **Step 4:**The system process query  **Step5:** System validates data  **Step6:** System updates tenants details in tenants file |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid Data Entered** |  | Step 6:The system displays an appropriate error message. |
|  |  | |
| **Conclusions** | Tenants details are changed | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Remove Tenant

Staff

Tenant

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Remove Tenant** | **Author:** |
| **Use Case Id** |  | **Date:** |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** | Tenant | |
| **Description** | This function Deletes tenant from the tenants file | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**Staff wants to remove Tenant from the System  **Step3:** Staff selects tenant from tenants list which will be removed from the system  **Step5:** Staff confirms to remove tenant from the system by clicking on save button | **Step 2:**System Displays GUI  **Step 4:** System retrieves Tenants data  **Step6:** System validate if specific tenant isn’t currently renting a property  **Step7:** System Removes tenant from the system |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Tenant is currently renting a property and cant be removed from the System** |  | **Step1:** System notify Staff that this tenant cant be removed from the System |
|  |  | |
| **Conclusions** |  | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### List Tenants

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **List Tenant** | **Author:** |
| **Use Case Id** |  | **Date:** |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** | Tenant | |
| **Description** | Staff members or admin can list tenants | |
| **Preconditions** | Tenants must be registered on the system before they can be listed | |
| **Trigger** |  | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**Staff member requests list of tenants  **Step 4:** | **Step 2:**System will retrieve tenants file and displays them in default order  **Step 3:** |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
|  |  | |
| **Conclusions** | A listing of all tenants in system will be generated | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Query tenants

## **Manage Staff**

### Add Staff

Staff

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add Staff** | **Author:** |
| **Use Case Id** |  | **Date:** |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | This process will add new staff member to system | |
| **Preconditions** |  | |
| **Trigger** | Properties file is requested | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Staff want add another staff member  **Step 3:** Staff enters others staff details:   * Forename * surname | **Step 2:**System displays GUI   * System add next Staff id   To enter Staff forename and surname   * System sets staff status to “Active” |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **A required field not entered** |  | **Step 1:** The system displays an appropriate error message |
|  |  | |
| **Conclusions** | A new property is added to the properties file | |
| **Post conditions** | Properties are now available for rent | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## **Manage Properties**

### Add Property

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add property** | **Author:** |
| **Use Case Id** |  | **Date:** |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | This process will add new property to system | |
| **Preconditions** |  | |
| **Trigger** | Properties files is requested | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Staff want add property  **Step 6:** Enter property details   * Street * Town * County * Type * Rent * noOfBeds * noOfBads * StaffID * OwnerId   **Step 7:** The user confirms that the property is to be added | **Step 2:** System retrieves Owner details from the Owners file  Step 3: The system retrieves details of allowed property types from the Types file  Step 4: The system determines the property Id  Step 5: The system displays on UI  **Step 8:** System validates the data entered:   * System add next id of property * System sets status to “Available” * All fields must be entered   **Step 9**: The system sets the property status to ‘available’  **Step 10:** The system saves the property details in the Properties file  Step 10: The system displays a confirmation message. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **A required field not entered** |  | **Step 1:** The system displays an appropriate error message |
|  |  | |
| **Conclusions** | A new property is added to the properties file | |
| **Post conditions** | Properties are now available for rent | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Amend Property

* 1. Amend Properties

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Amend property** | **Author:** |
| **Use Case Id** |  | **Date:** |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** | Staff | |
| **Description** | Staff needs to amend the details of a property. This requires the system to retrieve property details from the properties file to amend property. | |
| **Preconditions** |  | |
| **Trigger** | Properties files is requested | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Staff want to amend property  **Step 3:** Staff specifies which property is to be amended. The following criteria might be entered:   * Town * Price * Status * PropertyId   **Step 5:** Staff selects desired Property  **Step 7:** The user confirms that the property is to be added | **Step 2:** System displays UI  **Step 4:** System lists queried property(s)  **Step 6:** System displays UI of Selected property  Details displayed:   * Street * Town * County * Rent * Type * Status   **Step 8:** System validates the data entered:   * Same street   **Step 9:** The system saves the property details in the Properties file  **Step 10:** The system displays a confirmation message. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **A required field not entered** | Staff enter same address | **Step 1:** The system displays an appropriate error message |
|  |  | |
| **Conclusions** | A property is amended and saved to the properties file | |
| **Post conditions** | Properties are now available for rent | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## **Manage Staff**

### Add Staff

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add Staff** | **Author:** |
| **Use Case Id** |  | **Date:** |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | This process will add new Staff member to system | |
| **Preconditions** |  | |
| **Trigger** | Staff files is requested | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Staff want add new Staff member to the System  **Step3:** Enter staff details:  details(house/department, no of rooms, location, weekly rent price) | **Step 2: System assigns next StaffID** andDisplays the UI  staff(forname,surename)  **Step4:** System asks user to save details or exit without saving |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
|  |  | |
| **Conclusions** |  | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Remove Staff

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Remove staff** | **Author:** |
| **Use Case Id** |  | **Date:** |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | This process will remove existing staff from system | |
| **Preconditions** |  | |
| **Trigger** | Member staff want to remove existing staff member | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Staff wants another staff member  **Step 3:** Staff selects another staff member to be deleted  **Step5:** Staff confirms the removal | **Step 2:** System retrieves summary details of all staff in order of name from the Staff file and displays the GUI  **Step4: The** System retrieves full staff details and details of any properties managed by this staff from the Staff file and the Properties file r and displays on GUI.  **Step6: The** System sets the staff Status to ‘removed’ in the Staff file.  **Step 7**: The System displays a confirmation message  **Step 8:** The system resets the GUI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  | **Step1:** Staff cancels removing another staff member from the System by clicking on cancel button | **Step2:** selected staff member will stay saved in staff file |
|  |  | |
| **Conclusions** |  | |
| **Post conditions** | Staff will be removed from staff file | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## **Manage Rentals**

### Rent Property

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Rent Property** | **Author:** |
| **Use Case Id** |  | **Date:** |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | This process will Rent property to tenant | |
| **Preconditions** |  | |
| **Trigger** | Properties and tenants and Owners files is requested | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Staff wants to process new Property for rent  **Step 6: S**taff may specify details:   * Property Type * Location * No Beds   **Step 8** : The staff confirms the search  **Step 10:** Staff selects chosen property by customer by selecting on property row and confirms by clicking on “Next” button | **Step 2:** System assigns next RentalID  **Step 3:** The system retrieves details of property types from the PropertyTypes File and loads on UI  **Step 4:** The system retrieves details (only once) of each town for which properties are registered from the Properties file  **Step 5:** System loads number of beds (1-5).  **Step 7:** Displays the UI  **Step 9:** The system retrieves details of all available properties from the Properties file which satisfy the specified criteria and displays on the UI  **Step 11:**System retrieve details of selected property  **Step 12:** System displays selected properties full details in new GUI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
|  |  | |
| **Conclusions** |  | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Terminate Rental

### Record Payment

### Issue Late Notice

### Return Deposit

# **System Model(data flow diagram)**

## **DFD Elements**

External Entities:

Tenant

Owner

Data Stores:

Properties file

Tenants file

Owners file

Processes:

## **Level 0 DFD**

Property Details

Rental Request

Owner

Property

Rental System

Tenant

Lease

## **Level 1 DFD**

Tenants

External Entities:

Tenant

Staff

Data Stores:

Properties file

Tenants file

Processes:

(Staff member needs to amend tenant)

D1

Tenant’s files

P1

Amend

Tenant

Staff

Tenant Details

P2.4

Add Tenant

## **Level 2 DFD**

(rent property)

P2.1

Process rentals

D1

Properties files

Customer

Rent propety

D1

Tenants files

P2.2

Process tenants

P2.3

Process

Payments

Rent Property

P2.4

Add Tenant

# **Data Model**

County

County\_Code {pk}

Name

Property

PropertyID {pk}

Street

Town

Rent

NoBeds

NoBaths

Garden

Status

🡨 has a

PropertyType

TypeCode {pk}

Description

0 ..\*

1..\*

Is for 🡪

1

1

is for 🡪

Tenant

TenantId {pk}

Surname

Forename

Phone

PPS

Rental

RentId {pk}

DateFrom

DateTo

Deposit

Staff

StaffID {pk}

Surname

Forename

DOB Date

Status

0..\*

1

0..\*

1

owns a 🡪

Owner

ownerId {pk}

surname

forename

1

0..\*

🡨 Oversees

0..\*

1

Has a 🡪

# **Relational Schema**

Properties (PropertyID, Street, Town, County\_Code, Rent, typeCode, StaffId, Garden, Status, OwnerId)

Counties(County\_Code, Name)

Owners(OwnerId, Forename, Surname)

PropertyTypes(typeCode, Description)

Staff(StaffId, Forename, Surname)

Rentals(RentId, DateFrom, DateTo, Deposit, PropertyId, TenantId,);

Tenants(TennantId, Surname, Forname, Phone, PPS);

# **Database Schema**

**Schema:** <Schema Name>

**Relation:** <Counties>

Attributes:

County\_Code char(2)

Name varchar(15)

Primary Key : County\_Code

[Foreign Key :]

**Relation:** <Owners>

Attributes:

OwnerId numeric(3)

Forename varchar(10)

Surname varchar(10)

ownerTelNo varchar(15)

Primary Key : OwnerId

[Foreign Key : ]

**Relation:** <ProeprtyTypes>

Attributes:

TypeCode char(1)

Description varchar(10)

Primary Key : TypeCode

[Foreign Key :]

Relation: <Staff>

Attributes:

StaffID number(2)

Surname varchar(10)

Forename varchar(10)

Status varchar(1)

Primary Key : StaffId

[Foreign Key :]

Relation: <Properties>

Attributes:

PropertyID number(3)

Street varchar(20)

Town varchar(15)

County\_Code char(2)

Rent number(5)

TypeCode varchar(1)

StatusId varchar(1)

NoBeds number(1)

NoBaths number(1)

StaffId number(1)

OwnerId number(1)

Primary Key :

CONSTRAINT pk\_propertyId PRIMARY KEY (propertyId)

[Foreign Key :

CONSTRAINT fk\_Prop\_Counties FOREIGN KEY (County\_Code) REFERENCES Counties,

CONSTRAINT fk\_Prop\_Types FOREIGN KEY (TypeCode) REFERENCES PropertyTypes,

CONSTRAINT fk\_Prop\_Staff FOREIGN KEY (StaffId) REFERENCES Staff,

CONSTRAINT fk\_Prop\_Owners FOREIGN KEY (OwnerId) REFERENCES Owners);

]

Relation: <Rentals>

Attributes:

RentId numeric(5) ,

DateFrom date,

DateTo Date,

Deposit numeric(4),

PropertyId numeric(3),

TenantId numeric(3),

Primary Key :

CONSTRAINT pk\_rentId PRIMARY KEY (rentId)

[Foreign Key :

CONSTRAINT fk\_Rent\_Properties FOREIGN KEY (PropertyId) REFERENCES Properties,

CONSTRAINT fk\_Rent\_Tenants FOREIGN KEY (TenantId) REFERENCES Tenants);

]

Relation: <Tenants>

Attributes:

TenantId numeric(3),

Surnname varchar(10),

Forename varchar(10),

TelNo varchar(15),

ppsNum varchar(9),

Primary Key :

CONSTRAINT pk\_tenantId PRIMARY KEY (tenantId))

[Foreign Key :]

# **Program Specifications**

# **Conclusion**

# **References:**