CIS 250
Final Project Final Draft: For Peer Review

Rikki Nelson
May 30, 2011

Background Information

Michael Castro owns Starlight Apartments. He is an entrepreneurial developer that purchased the apartments in mid-construction, for very cheap. He intends to develop the property to realize full property value as well as receive continual income from rentals. Starlight Apartments has three buildings completed and available for rental operation. Each building has three floors, and each floor has four apartment units. The next four buildings are in finishing stage, while the remaining five buildings are still at initial construction. Parking is assigned; Uncovered is included with apartment rental, but a tenant may upgrade to a Covered Attached or Covered Adjacent space by paying an additional fee.

A system is desired to track all tenants, apartments, storage units, and parking units. A database is part of the solution because there may be more than one tenant renting an apartment, apartment availability depends on construction status and tenancy, storage units are assigned to apartments, and parking is assigned but depends on tenant rental agreement. Different reports are needed for different people.

System Requirements

- **Problem:** An automated database is required to track tenants, apartments, storage units, and parking spaces. The current system is entirely manual, which causes employees to be over-stressed as they struggle to keep up with changes and required updates.
- **People:**
  - Owner – Michael Castro
  - Manager – Jaymie Voss
  - Bookkeeper – Judy Smart
  - Tenants
- **Current Processes:**
  - After buildings are completed, units become available for rent.
  - Available units are considered vacant until rented.
  - Rented apartments are considered occupied.
  - Apartment units have differing sizes (studio, 1, 2 or 3 bedrooms) that affect rental price.
  - Apartments may have differing amenities that affect rental price.
  - Occupied units may have more than one tenant renting if 2 or more bedrooms.

- Storage units in buildings are assigned to apartment units.
- Each tenant will have one parking space assigned to them.
- Uncovered parking is assigned to each tenant on a first come-first serve basis.
- A tenant may pay additional fee with monthly rent for covered attached or covered adjacent parking.
- Covered parking is an upgrade, therefore a tenant waives right to free uncovered parking space when renting a covered space.
- All newly occupied units must have move in dates.
- All occupied units must have an estimated move-out date.
- All newly vacated units must have dates vacated.
-

❖ **Information Requirements:**

- Existing Data include:
  - Tenant Information: Personal information, Apartment Unit, Parking Space, Monthly Rent, Estimated Move-In Date, and Estimated Move-out Date.
  - Apartment Information: Building Number, Apartment Number, Size, Amenities, Storage Unit Assigned, Rental Price, Status, and date of most recent status.
  - Parking: Type, Space Number, Tenant Assigned, Rental Price, and Status.
- Existing Inputs include:
  - Prospective Tenant records, to include expected move-in date.
  - Availability of newly constructed buildings/units.
  - Change in apartment occupancy or amenities.
  - Change in parking space assignment.
- Existing Outputs include:
  - Tenant Report (tenant, unit, parking, move-in, and move-out dates)
  - Apartment Availability
  - Parking Space Availability
  - Revenue
  - Tenant Monthly Bill

❖ **Strengths:**

- Data is in tabular form in Excel (keeps records together).

❖ **Weaknesses:**

- Update of data must involve multiple locations (duplicated data).
- Different data tables don't easily relate fields and records for various reports.
- Reports must be customized and reprinted after every update.

- **Objectives –** the new proposed system is intended to provide/solve the following:
  - System:
    - Validation of data.
    - Relation of data entities and attributes for quicker, easier updating.
  - Inputs:
    - Ease of updating apartment and parking status.
    - Archiving of Former Tenants.
  - Outputs:
    - Rental Unit Availability Report
    - Rental Revenue Report.
    - Tenant Rental Pricing Addendum (easy to use form to attach to standard Lease Agreement.
    - Tenant Monthly Rent Due billing letter.
- **Benefits –** what are the benefits the new system will provide to the business?
  - Quick, easy update of records and fields.
  - Quick and easy reports that can be read on computer screen as well as printed.
  - Complete data integrity.
- **Alternative Solutions:**
  - Provide Judy training. Excel 2007 and 2010 have the capability of linking and updating tables to improve update, tracking, and creation of some forms utilizing macros. Much of Judy Smart's issues could be resolved by restructuring the excel tables she uses. Also, adding all buildings/apartments to the excel data will make it easier to update status rather than adding new records each time a building is finished. The latter would then allow for keeping a status and making simple changes when construction is completed.

Purpose & Scope

The purpose of the Starlight Apartments database is to provide an automated system that will accurately maintain data that supports the rental of residential property to our tenants. The scope of this project involves tracking the features and availability of apartments, parking spaces, and storage rooms. The scope also includes tracking tenant information and rental records. The scope of this project does not include processing tenant application or credit management, tracking employees, tracking maintenance or repair activity, processing accounts receivable accounting (tenant rent), processing payroll accounting, or processing Profit/Loss accounting.

Information Requirements

- Rental Records
  - PK: Rental Record ID
  - Rental start date
  - Rental terms (lease or month-to-month)
  - Duration (legal length of lease or expected number of months)
- Tenants:
  - PK: Tenant ID
  - Personal information
  - Contact information
- Parking Spaces:
  - PK: Parking ID
  - Parking Category
  - Parking Price
  - Space Number
- Apartments:
  - PK: Apartment Number
  - Bedrooms (number of)
  - Base Price (based on number of bedrooms)
  - Baths (number of)
- Features:
  - PK: Feature Id
  - Feature description
  - Feature price
- Buildings
  - PK: Building ID
  - Building Name
  - Building Designation
- Storage Rooms
  - PK: Storage ID
  - Storage room number

Business Rules

Data

- There are (12*4*3=) 144 apartments in the complex.
- It is possible for one apartment to house many tenants (1-to-many relationship).
- Each apartment unit has designated a single storage unit (1-to1 relationship).
- Monthly rent billed will depend on apartment features and additional charges for upgraded parking (thus is a calculated form).

- All tenants will be assigned an uncovered parking space (1-to-1 relationship) for free; assignments are all first come – first served basis.
- Many apartments may have many of the same features (many-to-many relationship), but each apartment will only have one specific combination of features (changed to 1-to-1 relationship).
-

  Process

- There are 12 buildings; as each completes construction then availability status for the apartments must be updated.
- There are 4 apartments/floor and 3 floors in each building; each must track availability status.
- All units, including those still under construction, must exist within the database to allow easier real-time updates.
- Manager must be able to view Availability and Occupancy after every update.

Issues

- Number of bathrooms doesn't add to price of the apartment unit? This is the case, but is contrary to current policy based on spreadsheet.

Assumptions

- A tenant is not entered into database until a tenant lease is signed. This is feasible as long as tenancy application remains outsourced.
- Only one tenant may rent a single bedroom.
- For studio and 1-bedroom apartments, only 1 tenant (or a married, childless couple) may rent a unit.
- For 2- and 3-bedroom apartments, there may be up to 2 or 3 separate tenants each (or a family unit of parents and children).
- If a tenant rents one or more covered parking spaces, then they forfeit their free, assigned, uncovered parking space.
- A tenant *may* rent more than one covered parking space.
- Storage Rooms are physically located within the same building to which it is assigned.
- Parking Spaces are located in the immediate vicinity of the buildings (there are no lots that are not adjacent to a building).
- Features and amenities are synonymous.
- Features are static—they will not change as modeled.

Final Table List

Name (conceptual)	Type	Description
Buildings	Data	The record of each building and its location.
Feature	Data	The record of all features that an apartment unit may have and its additional price.
Parking Space	Data	The record of all parking spaces and which building each is assigned to. Type of space, rental amount, and availability status are also important.
Parking Type	Validation	Validates the type that a parking space is as well as provides look-up of price.
Rental Parking	Linking	This links the 0, 1, or more parking spaces that may have 0, 1, or more rental records.
Rental Record	Data	The records of each tenant lease with the associated apartment unit, parking spaces, and storage room. The type of lease, monthly charge, move-in and termination dates are of most importance to optimize rental revenue. This is the central table.
Rental Tenant	Linking	This links 1 or more tenants that may be have 0, 1, or more rental records.
Storage Unit	Data	The record of all storage rooms.
Tenant	Data	The personal information of each tenant and their status.
Unit	Data	The record of all apartment units and number of bathrooms.
Unit Feature	Linking	This links the 0, 1, or more features that may exist in 0, 1, or more apartment units.
Unit Type	Validation	Validates the type of apartment based on the number of bedrooms as well as provides look-up of price.

List of Attributes

BUILDING:
BUILDING_ID, Building_Number, Building_Name.

FEATURE:
FEATURE_ID, Feature_Description, Feature_Price.

PARKING_SPACE:
PARKING_ID, Parking_Space_Num, *PARKING_TYPE_ID*, *BUILDING_ID*.

PARKING_TYPE:
PARKING_TYPE_ID, Park_Type_Category, Park_Category_Price.

RENTAL_PARKING:
PARKING_ID, RENTAL_ID.

RENTAL_RECORD:
RENTAL_ID, Rental_Start_Date, Rental_Terms, Rental_Duration,
STORAGE_ID, *UNIT_ID*.

RENTAL_TENANT:
TENANT_ID, *RENTAL_ID*.

STORAGE_UNIT:
STORAGE_ID, Storage_Unit_Number, *BUILDING_ID*.

TENANT:
TENANT_ID, Tenant_Last_Name, Tenant_First_Name, Tenant_DOB,
Tenant_Gender, Tenant_Contact, Tenant_Email.

UNIT:
UNIT_ID, Unit_Number, Unit_Baths, *UNIT_TYPE_ID*, *BUILDING_ID*.

UNIT_FEATURE:
UNIT_ID, *FEATURE_ID*.

UNIT_TYPE:
UNIT_TYPE_ID, Unit_Bedrooms, Unit_Bedrooms_Price.

Conceptual Model Assertions

A Building may relate to 0, 1, or more than one Parking Space; however, each Parking Space must relate to one and only one Building.

A Building may relate to 0, 1, or more than one Storage Unit; however, each Storage Unit must relate to one and only one Building.

A Building may relate to 0, 1, or more than one Unit; however, each Unit must relate to one and only one Building.

A Feature may relate to 1 or more than one Unit Feature; however, each Unit Feature must relate to one and only one Feature.

A Parking Space may be related to 1 or more than one Rental Parking; however, each Rental Parking must relate to one and only one Parking Space.

A Parking Type may relate to 0, 1, or more than one Parking Space; however, each Parking Space must relate to one and only one Parking Type.

A Rental Record may be related to 1 or more than one Rental Parking; however, each Rental Parking must relate to one and only one Rental Record.

A Rental Record may be related to 1 or more than one Rental Tenant; however, each Rental Tenant must relate to one and only one Rental Record.

A Storage Unit may relate to 0, 1, or more than one Rental Record; however, each Rental Record must relate to one and only one Storage Unit.

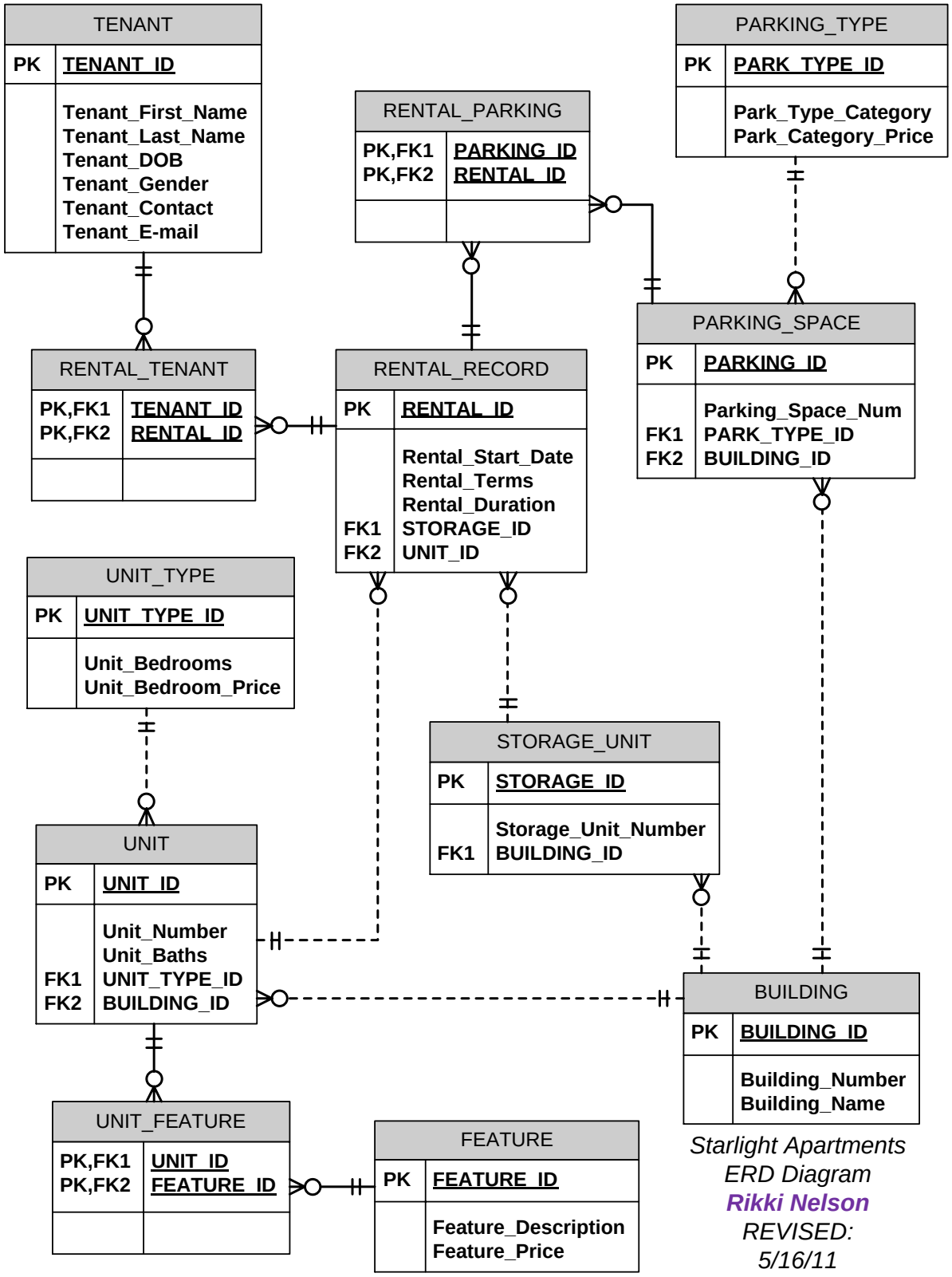
A Tenant may be related to 1 or more than one Rental Tenant; however, each Rental Tenant must relate to one and only one Tenant.

A Unit may relate to 1 or more than one Unit Feature; however, each Unit Feature must relate to one and only one Unit.

A Unit may relate to 0, 1, or more than one Rental Record; however, each Rental Record must relate to one and only one Unit.

A Unit Type may relate to 0, 1, or more than one Unit; however, each Unit must relate to one and only one Unit Type.

Logical-Level Entity Relationship Diagram



Starlight Apartments
ERD Diagram
Rikki Nelson
REVISED:
5/16/11

Meta Data Dictionary

Entity Name	Attribute Name	Definition	Domain Constraints	Referential Integrity Restraints
BUILDING	BUILDING_ID	Unique identifier for each building.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Building_Number	The number that labels the building.	Required, Non-unique, Char (6).	
	Building_Name	The name that each building is referred as.		
FEATURE	FEATURE_ID	Unique identifier for each feature.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Feature_Description	A descriptive title of each feature.	Required, Non-unique, Char (25).	
	Feature_Price	The additional price charged for each feature.	Required, Non-unique, Numeric(10).	
PARKING_SPACE	PARKING_ID	Unique identifier for each parking space.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Parking_Space_Num	The number that labels the parking space.	Required, Non-unique, Char (6).	
	PARKING_TYPE_ID		Required. **FK** (PARKING_TYPE)	A parking space must have a parking type.
	BUILDING_ID		Required. **FK** (BUILDING)	A parking space must be associated with a building.
PARKING_TYPE	PARKING_TYPE_ID	Unique identifier for each type of parking space.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Park_Type_Category	The descriptive name for the type of parking space.	Required, Non-unique, Char (10).	Valid entries: Uncovered, Attached, or Adjacent.
	Park_Category_Price	The additional price charged for the parking.	Required, Non-unique, Numeric (10).	

RENTAL_PARKING	PARKING_ID		Required. **FK** (PARKING_SPACE)	**CPK** (Composite Primary Key)
	RENTAL_ID		Required. **FK** (RENTAL_RECORD)	**CPK** (Composite Primary Key)
RENTAL_RECORD	RENTAL_ID	Unique identifier for each record of a unit rental.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Rental_Start_Date	The date the rental becomes effective.	Required, Non-unique, Numeric(8).	
	Rental_Terms	The type of rental in accordance with State law.	Required, Non-unique, Char (10).	Valid entries: Lease, or Monthly
	Rental_Duration	The number of months of rental.	Required, Non-unique, Numeric(3).	
	STORAGE_ID		Required. **FK** (STORAGE_UNIT)	A rental record must include a storage unit.
	UNIT_ID		Required. **FK** (UNIT)	A rental record doesn't exist without a unit.
RENTAL_TENANT	TENANT_ID		Required. **FK** (TENANT)	**CPK** (Composite Primary Key)
	RENTAL_ID		Required. **FK** (RENTAL_RECORD)	**CPK** (Composite Primary Key)
STORAGE_UNIT	STORAGE_ID	Unique identifier for each storage unit.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Storage_Unit_Number	The number that labels the storage unit.	Required, Non-unique, Char (6).	
	BUILDING_ID		Required. **FK** (BUILDING)	A storage unit doesn't exist without a building.

TENANT	TENANT_ID	Unique identifier for each tenant.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Tenant_Last_Name	Tenant's last name.	Required, Non-unique, Char (25).	
	Tenant_First_Name	Tenant's first name.	Required, Non-unique, Char (25).	
	Tenant_DOB	Tenant's date of birth.	Required, Non-unique, Numeric(8).	
	Tenant_Gender	Tenant's gender.	Required, Non-unique, Char (1).	
	Tenant_Contact	Tenant's contact phone number.	Required, Non-unique, Numeric (10).	
	Tenant_Email	Tenant's email address.	Optional, Unique, Char (25).	
UNIT	UNIT_ID	Unique identifier for each apartment unit.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Unit_Number	The number that labels the apartment unit.	Required, Non-unique, Char (6).	
	Unit_Baths	The number of bathrooms that the apartment unit has.	Required, Non-unique, Numeric(3).	Valid entries: 1, 1.5, 2, or 2.5.
	UNIT_TYPE_ID		Required. **FK** (UNIT_TYPE)	A unit must have a unit type.
	BUILDING_ID		Required. **FK** (BUILDING)	A unit doesn't exist without a building.
UNIT_FEATURE	UNIT_ID		Required. **FK** (UNIT)	**CPK** (Composite Primary Key)
	FEATURE_ID		Required. **FK** (FEATURE)	**CPK** (Composite Primary Key)

UNIT_TYPE	UNIT_TYPE_ID	Unique identifier for each type of apartment unit.	System Assigned. Unique. Numeric (10)	**PK** (Primary Key)
	Unit_Bedrooms	The number of bedrooms a unit has.	Required, Non-unique, Numeric(1).	Studio apartments have 0 bedrooms.
	Unit_Bedroom_Price	The base price to be charged for rental of each unit type.	Required, Non-unique, Numeric(10).	