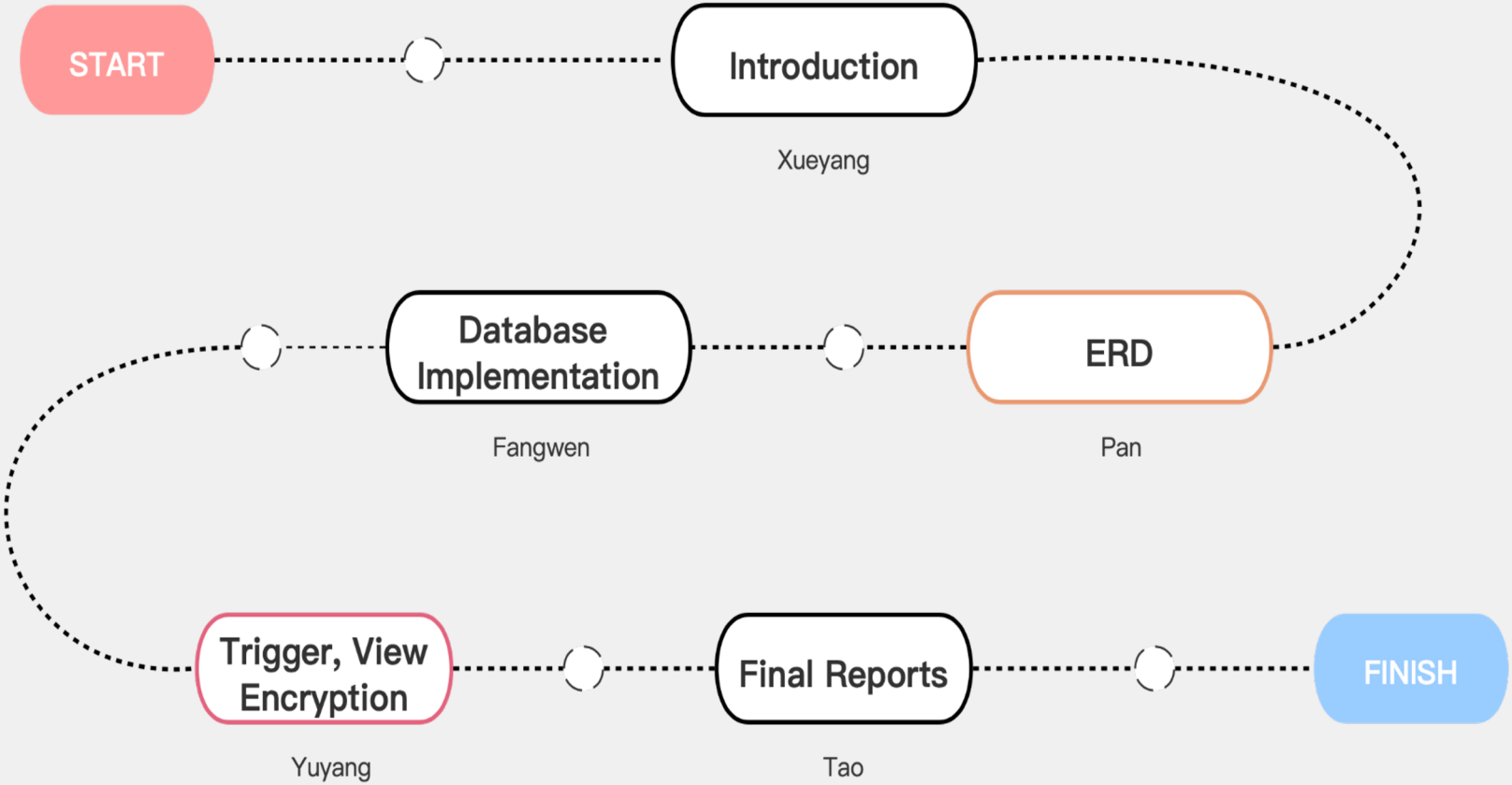
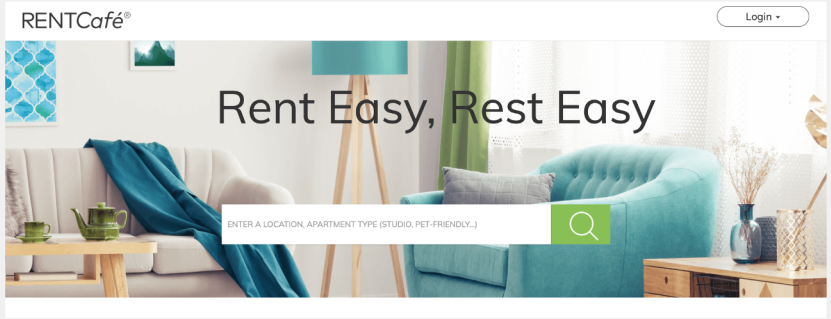
Online Apartment Management System

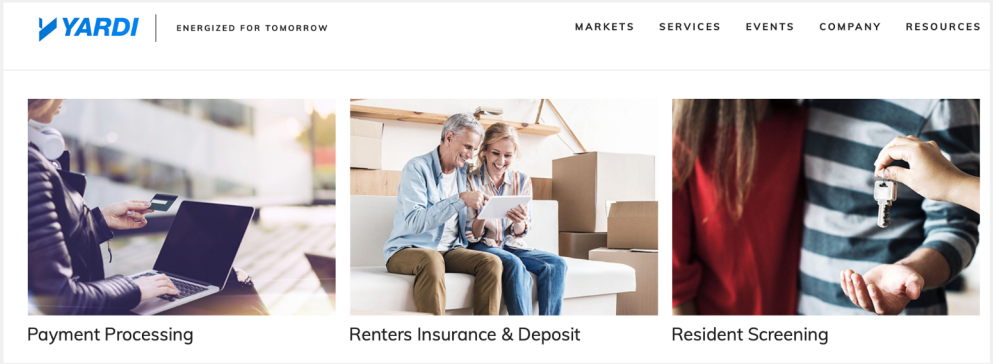
DAMG6210 Dec 2021
Group 5
Fangwen Ge, Yuyang Han, Lin Tao
Xueyang Li, Pan Xu

Our Story


START
Introduction
Xueyang
ERD
Pan
Database Implementation
Fangwen
Trigger, View Encryption
Yuyang
Final Reports
Tao
FINISH

Introduction





Users

Renters,
Apartment Managers,
Maintenance Workers...

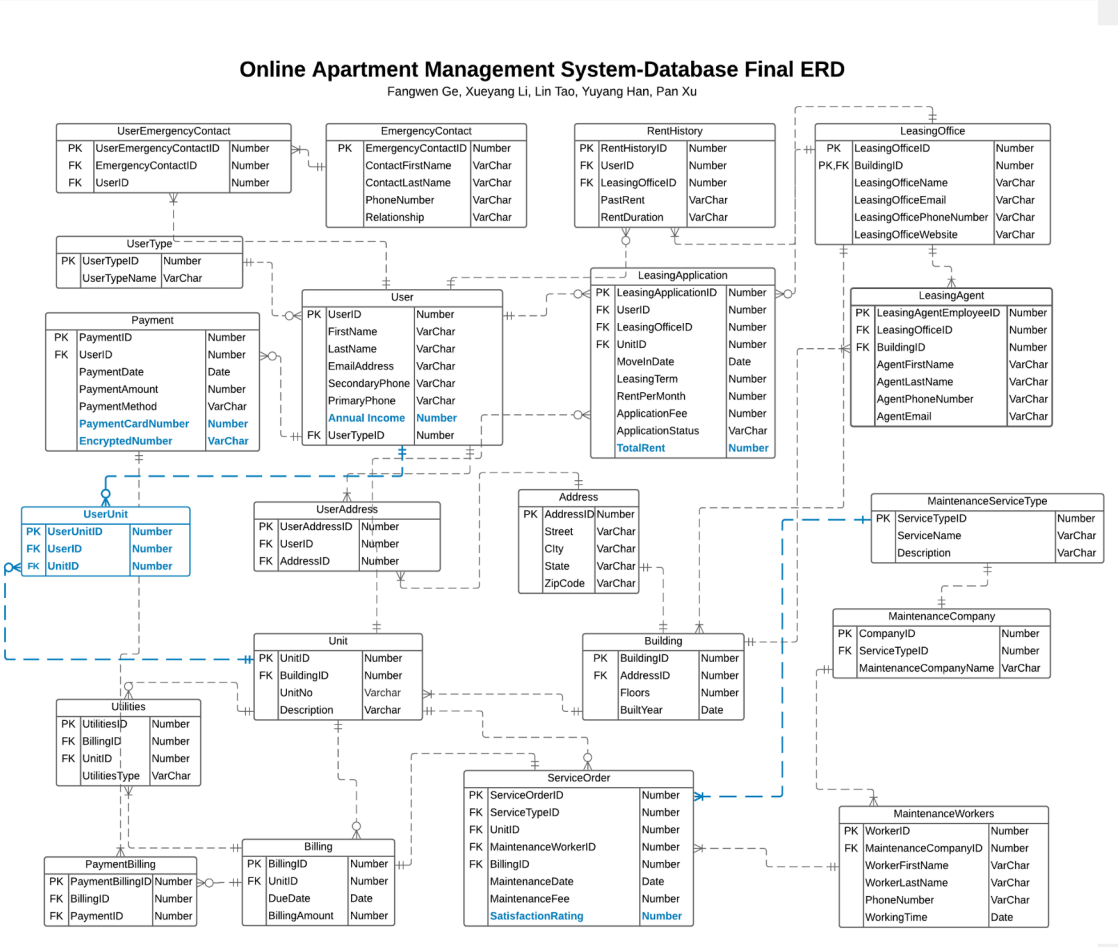
Objectives

Maintain data
Perform searches
Track status

Entities

User, Building, Leasing office, Billing, Utilities...

ER Diagram



Users

Tenants (potential applicants)
Leasing
Maintenance Workers

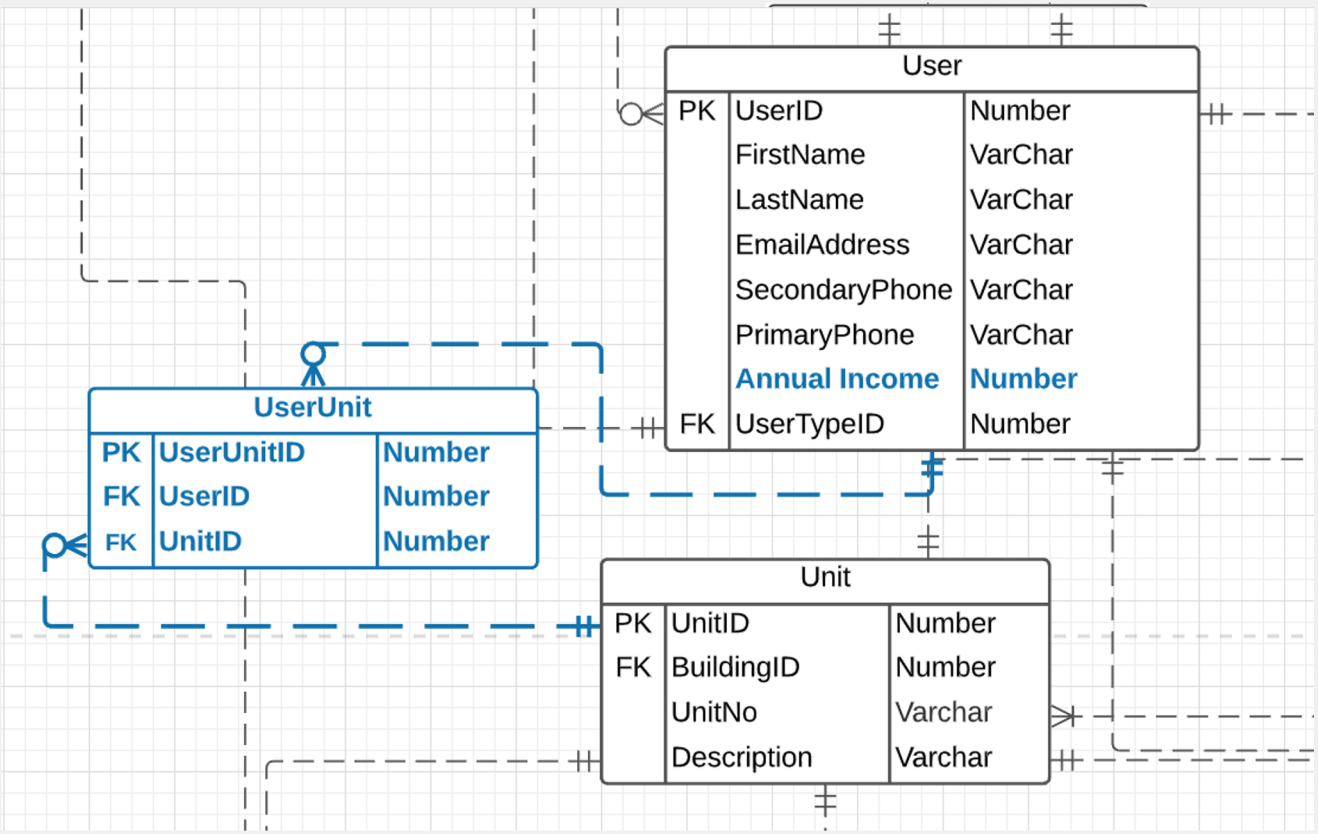
Building & Office

Operation process
Segmentation
Buildings and Units

Maintenance & Service

Orders
Track
Utilities

Increased Details & Entity



Database Implementation

- User Table
- Users can be of different types, as defined in the UserType table

```
Create TABLE [User]
(UserID INT IDENTITY PRIMARY KEY NOT NULL,
FirstName Varchar(30),
LastName VarChar(30),
EmailAddress Varchar(50),
PrimaryPhone Varchar(50),
SecondaryPhone Varchar(50),
AnnualIncome FLOAT,
UserTypeID INT);
```

Database Implementation

- Resolving many-to-many relationships using linking table and surrogate keys]
- Non-identifying relationship

```
Create TABLE UserAddress(
UserAddressID INT IDENTITY PRIMARY KEY NOT NULL,
UserID INT references [User](UserID),
AddressID INT references Address(AddressID));

CREATE TABLE UserUnit
(UserUnitID INT Identity Primary Key NOT NULL,
 UserID INT REFERENCES [User](UserID),
 UnitID INT REFERENCES Unit(UnitID)
  );

 CREATE TABLE UserEmergencyContact
(UserEmergencyContactID INT Identity Primary Key NOT NULL,
 EmergencyContactID INT REFERENCES EmergencyContact(EmergencyContactID),
 UserID INT REFERENCES [User](UserID)
  );
```

Database Implementation
- Constraints and Computed Column

- Function `checkLeasingTerm`
  - Checks that an application must have a leasing term of more than 9 months
  - Added to check constraint `RejectShortLease`
- Computed column
  - Total rent as `LeasingTerm * RentPerMonth`

```
create function checkLeasingTerm (@LeasingTerm int)
returns int
as
begin
    declare @result int = 0;
    if @LeasingTerm > 9
    begin
        set @result = 1;
    end;
    return @result;
end

go
alter table LeasingApplication add constraint
RejectShortLease check (dbo.checkLeasingTerm(LeasingTerm) = 1);
```

```
alter table LeasingApplication add
    TotoalRent as LeasingTerm * RentPerMonth;
```

Database Implementation - Insert Data

- We inserted data into the database using varieties of methods
  - Manual insert statement for basic information (UserType)
  - Randomized data insertion using Javascript and Excel import wizard for more complex data

```
const firstName = [
    'Liam', 'Olivia',
    'Noah', 'Emma',
    'Oliver',   'Ava',
    'Elijah',   'Charlotte',
    'William',  'Sophia',
    'James',    'Amelia',
    'Benjamin', 'Isabella',
    'Lucas',    'Mia',
    'Henry',    'Evelyn',
    'Alexander',    'Harper'
]

const lastName = ["Anderson", "Ashwoon", "Aikin", "Bateman", "Bongard", "Bowers", "Boyd", "Cannon", "Cast", "Deitz", "Dewalt", "Eb

for (let i = 0; i < 100; i++) {
    const first = firstName[Math.floor(Math.random() * firstName.length)];
    const last = lastName[Math.floor(Math.random() * lastName.length)];
    const email = first+last+Math.floor(Math.random() * 10) + '@gmail.com';
    const primaryPhone = createRandomPhoneNumber();
    const secondaryPhone = createRandomPhoneNumber();
    let AnnualIncome = Math.random() * (50000) + 50000;
    AnnualIncome = AnnualIncome.toFixed(2)
    const userTypeID = Math.floor(Math.random() * 9) + 1 + 5;
    console.log(`('${first}', '${last}', '${email}', '${primaryPhone}', '${secondaryPhone}', ${AnnualIncome}, ${userTypeID}), `);
}

function createRandomPhoneNumber() {
    let phoneNumber = '';
    for (let i = 0; i < 10; i++) {
        phoneNumber += Math.floor(Math.random() * 10);
    }

    return phoneNumber
}
```

Trigger

```
CREATE TRIGGER BillingAfterMovein ON Billing
AFTER INSERT, UPDATE
AS
  IF EXISTS
      (SELECT 'TRUE'
       FROM INSERTED i
       JOIN LeasingApplication la
       ON i.UnitID= la.UnitID
       WHERE i.DueDate < la.MoveInDate
        )
      BEGIN
      ROLLBACK TRANSACTION
      RAISERROR ('The billing date should be later than the movein date.',16,1)
      END

ALTER TABLE Billing ADD CONSTRAINT BillingAfterMovein CHECK
(dbo.CompareDueDate(UnitID)=1);
```

- Billing after move-in

123 UnitID	MoveInDate
10	2020-05-18
19	2021-04-13
27	2019-09-26
330	2020-04-22

```
INSERT Billing Values(10,'2020-01-01',1700);
```

SQL Error [50000] [S0001]: The billing date should be later than the movein date.

Details >>

Encryption

```
CREATE MASTER KEY
ENCRYPTION BY PASSWORD = 'group5';
-- Create certificate to protect symmetric key
CREATE CERTIFICATE aptCardnumber
WITH SUBJECT = 'apt Test Certificate',
EXPIRY_DATE = '2022-10-31';
-- Create symmetric key to encrypt data
CREATE SYMMETRIC KEY aptCardnumberKey
WITH ALGORITHM = AES_128
ENCRYPTION BY CERTIFICATE aptCardnumber;
ALTER TABLE Payment
      ADD EncryptedNumber Varbinary(128);
GO
OPEN SYMMETRIC KEY aptCardnumberKey
DECRYPTION BY CERTIFICATE aptCardnumber;
UPDATE Payment
SET EncryptedNumber =EncryptByKey(Key_GUID('aptCardnumberKey'),
CAST(PaymentCardNumber AS VARCHAR(128)));
```

- Payment Card Number Encryption

PaymentID	UserID	PaymentDate	PaymentAmount	PaymentMethod	EncryptedNumber
1	178	2022-01-02	3,000	CHECK	æÇ Q¦J¡ £Ù û P Å® t ¼ ¡b+ÊV... [52]
2	181	2022-01-04	720	CASH	æÇ Q¦J¡ £Ù û P ï£< ¾Q; ²À¿ ... [52]
3	101	2022-01-10	2,000	ONLINEPAYMENTS	æÇ Q¦J¡ £Ù û P ´Kf Ùi ú^· ... [52]
4	102	2022-01-05	1,480	ONLINEPAYMENTS	æÇ Q¦J¡ £Ù û P k¯ù . ¨ Elk ... [52]
5	103	2022-01-06	2,153	ONLINEPAYMENTS	æÇ Q¦J¡ £Ù û P ê ú?«[à´ê... [52]
6	104	2022-01-07	1,569	ONLINEPAYMENTS	æÇ Q¦J¡ £Ù û P ÷ (ü¤ é e ... [52]
7	105	2022-01-02	2,233	ONLINEPAYMENTS	æÇ Q¦J¡ £Ù û P ÁJ Þ ÖÌliõ ... [52]
8	106	2022-01-02	1,300	ONLINEPAYMENTS	æÇ Q¦J¡ £Ù û P ~ c>$ÓJÙÍ ... [52]
9	107	2022-01-09	1,892	ONLINEPAYMENTS	æÇ Q¦J¡ £Ù û P ÂO ¡X. Õ~í è... [52]
10	108	2022-01-04	1,935	CHECK	æÇ Q¦J¡ £Ù û P Aþxd. y C± ... [52]
11	109	2022-01-02	1,379	CHECK	æÇ Q¦J¡ £Ù û P ¯8 ¨óL× (¾... [52]
12	179	2022-01-07	1,400	ONLINEPAYMENTS	æÇ Q¦J¡ £Ù û P Æ ÀN <ïkç:... [52]
13	182	2022-01-07	1,400	ONLINEPAYMENTS	æÇ Q¦J¡ £Ù û P C¶?PA¤ (... [52]
14	180	2022-01-02	1,300	ONLINEPAYMENTS	æÇ Q¦J¡ £Ù û P `Ö'¤8 Wk*/$y... [52]
15	183	2022-01-05	1,342	ONLINEPAYMENTS	æÇ Q¦J¡ £Ù û P ¹f ô §u1SýÅ±... [52]
16	185	2022-01-05	3,458	ONLINEPAYMENTS	æÇ Q¦J¡ £Ù û P ½Dç 0 q¢ yÿâ... [52]
17	188	2022-01-07	2,804	ONLINEPAYMENTS	æÇ Q¦J¡ £Ù û P ç ,ô Î Df4... [52]

Views

```
CREATE VIEW  MaxBillingInEachBuilding
AS
WITH temp AS(
SELECT u.BuildingID, u.UnitNo,SUM(b.BillingAmount)[Max Billing],
    RANK() OVER(PARTITION BY u.BuildingID ORDER BY SUM(b.BillingAmount) D
   FROM Unit u JOIN Billing b ON u.UnitID =b.UnitID
   GROUP BY u.BuildingID, u.UnitNo)
SELECT BuildingID, UnitNo,[Max Billing]
FROM temp
WHERE [rank]=1;
```

123 BuildingID	ABC UnitNo	123 Max Billing
1	103	3,720
2	201	2,642
3	409	3,458
4	401	2,804
5	309	3,656
6	205	2,652
7	502	2,399

```
CREATE VIEW UnitMember
AS
WITH temp
AS(
   SELECT ut.UnitID , CONCAT(u.FirstName,' ',u.LastName ) [Name]
    FROM UserUnit uu
   JOIN [User] u ON uu.UserID =u.UserID
   JOIN Unit ut ON ut.UnitID =uu.UnitID
   )
 SELECT DISTINCT UnitID , STUFF((SELECT ','+ CAST([Name] AS CHAR
                                FROM temp t1
                                WHERE t1.UnitID =t2.UnitID
                                FOR XML PATH('')) , 1, 1, '
FROM temp t2;
```

123 UnitID	ABC Names		
3	Benjamin Trusela	,Alexander Wakefield	,Oliver Zeller
10	Olivia Schickowski		
17	Mia Woo		
19	William Kalleg		
20	Alexander Davis		
21	Amelia Deitz		
22	Olivia Schutz		
24	Benjamin Stahl		
26	Elijah Resnick		
27	Olivia Duffman		
34	Isabella Reamer	,Emma Doran	
72	Alexander Tapia	,James Hoffman	
180	Liam Orwig	,Amelia Weinstein	,Liam Cast
222	Evelyn Lawicki	,Sophia Duffman	
280	Evelyn Myers	,Amelia Dugelman	,William Tho
314	Henry Vader	,Amelia Christensen	
316	James Cannon	,Lucas Sweeney	

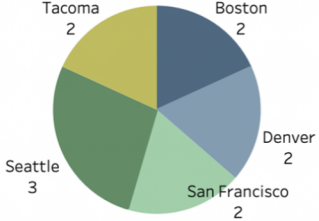
```
CREATE View totalBillingByMonth
AS
SELECT UNITID,SUM(BillingAmount)[ttlamount]
FROM Billing
GROUP BY UNITID,Year(DueDate),Month(DueDate)
```

123 UNITID	123 ttlamount
3	3,720
10	2,095
17	1,480
19	2,153
20	1,569
21	2,233
22	1,481
24	1,892
26	1,935
27	1,379
34	2,981
72	2,642
180	3,458
222	2,804

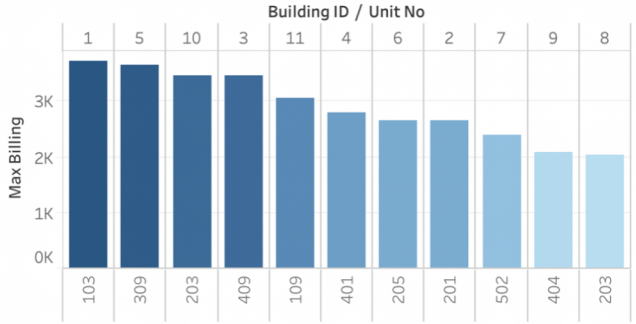
Reports

Apartment Location

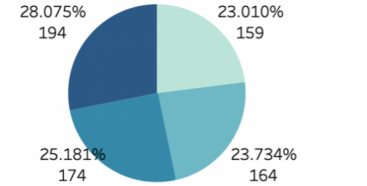
Number of Apartments in Each City



Maximum Billing Unit in Each Building



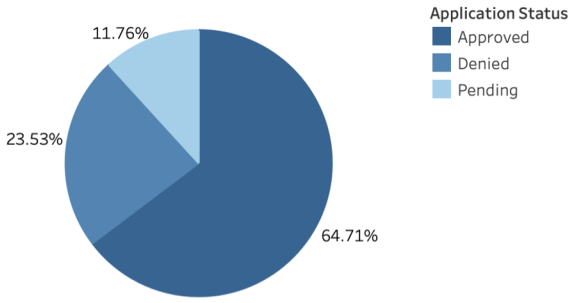
Percentage of Four Unit Types



Highlight D.. No items highlighted

Reports

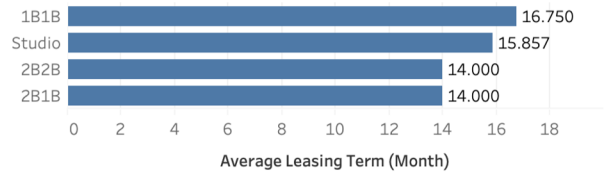
Application Status(%)



Approval rate: 73%

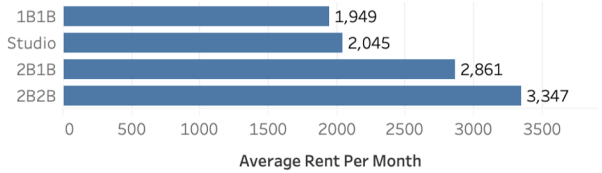
Leasing..	Leasing Office Name	Application Status: Approved	Application Status: Denied	Application Status: Pending
1	EquityResidential	○	○	
2	EquityResidential	○		
3	Onni Residential	○	○	
4	Avenue5 Residential	○		
5	Palladium Real Estates	○	○	
6	Greystar	○		○
7	Purple Properties	○		
8	Fantastic Estates	○		
9	The Perfect Tenant	○		
10	Golden Real Estates	○		
11	Fluffy Homes	○		○

Avg Leasing Term for different unit types



Overall Avg Leassing Term: 15

Avg Rent Per Month For Different unit types



Overall Avg Rent Per Month: $2425

Thank you!

Questions?