

Simple Real Estate

Attribute Driven Design

Authors: Nicholas Boland - Craig Lillemoen

Contents

1 ADD Step 1: Inputs	1
2 Iterations	2
2.1 Iteration 1	2
2.1.1 ADD Step 2	2
2.1.2 ADD Step 3	3
2.1.3 ADD Step 4	3
2.1.4 ADD Step 5	3
2.1.5 ADD Step 6	3
2.1.6 ADD Step 7	4
2.2 Appendix A	4

1 ADD Step 1: Inputs

- *Design Purpose:* This is a greenfield system in a relatively known domain. The organization will perform development following an agile process with short iteration cycles to get feedback fast and mitigate the risks. An architectural design is required to make conscious decisions, satisfy architectural drivers, and avoid future rework.

- *Functional requirements:*

- UC1: Add/Remove Properties
- UC2: View/Modify Properties
- UC3: Add/Remove Units
- UC4: View/Modify Units
- UC5: Add Maintenance Records and Quotes
- UC6: Add Expenses
- UC7: Generate Profit/Expense Reports
- UC8: Add/Remove Tenants
- UC9: View/Modify Tenants
- UC10: Add Rental Agreements

- UC11: View/Modify Rental Agreements
- *Most Relevant Use-cases:*
 1. UC1, UC2, UC3, UC4, UC8, UC9, UC10
- *Quality Attributes:*
 - QA1: Security
 - QA2: Usability
 - QA3: Manageability
 - QA4: Modularity
- *Architectural Constraints:*
 - CON1: The application must be accessed via a web browser
 - CON2: User must have an active network connection
 - CON3: Prior reports must be stored and accessible
 - CON4: Modular design allowing future modifications with ease
- *Architectural Concerns:*
 - CNR1: Developing a ground up system
 - CNR2: Leveraging team's knowledge in Python, Django, HTML, and SQLite
 - CNR3: Work allocated to small development team

2 Iterations

2.1 Iteration 1

2.1.1 ADD Step 2

1. *Iteration Goal:* Design initial Django website framework to create properties, units, and rental agreements pages
2. *Design Drivers:*
 - (a) CNR1: Developing a ground up system
 - (b) CON1: The application must be accessed via a web browser
 - (c) QA2: Usability
 - (d) QA4: Modularity
 - (e) UC1: Add/Remove Properties
 - (f) UC2: View/Modify Properties
 - (g) UC3: Add/Remove Units
 - (h) UC4: View/Modify Units
 - (i) UC10: Add Rental Agreements
 - (j) UC11: View/Modify Rental Agreements

2.1.2 ADD Step 3

1. *Elements Chosen to Decompose/Refine:*

- (a) Ground Up
 - i. As this is greenfield development, and we are in the initial iteration, the element to refine is the entire system.
- (b) Element 2

2.1.3 ADD Step 4

1. *Design Concepts:*

- (a) Concept 1: Design using Django's Model-View-Template (MVT) Architecture.
 - i. This concept was chosen as it is allowed for in-built ease of use, and a clearly defined separation of models and views.
- (b) Concept 2: Design using a Client-Server Architecture.
 - i. This concept was not directly chosen, but is indirectly implemented by using the Django framework.

2.1.4 ADD Step 5

1. Models - Work Allocated to Nicholas

- (a) 1 Model for Property
- (b) 1 Model for Tenant
- (c) 2 Models for Rental Agreement and Rental Invoice
- (d) 2 Models for Maintenance Record and Maintenance Record Item
- (e) 3 Models for Maintenance Quote, Maintenance Quote Item, and Maintenance Quote Invoice
- (f) 2 Models for Expense Record and Expense Record Item

2. Views - Work Allocated to Craig

- (a) 3 Views and 2 Templates for Property
- (b) 3 Views and 2 Templates for Unit
- (c) 3 Views and 2 Templates for Tenant
- (d) 3 Views and 3 Templates for Rental Agreement

2.1.5 ADD Step 6

After Designing the following three pages, we had decided that for every view, that for every instance of the objects would be separated by some boxes signifying the difference between each instance. See figures in Appendix A.

2.1.6 ADD Step 7

1. *Analysis:*

- (a) CNR1: Developing a ground up system
 - i. This concern was partially addressed as the initial framework was designed
- (b) CON1: The application must be accessed via a web-browser
 - i. This Constraint was addressed by implementing the Django MVT Architecture
- (c) QA2: Usability
 - i. Usability was not addressed in the initial iteration, as functionality was the goal.
- (d) QA4: Modularity
 - i. Modularity is continually being addressed as new design decisions are made.
- (e) UC1: Add/Remove Properties, UC2: View/Modify Properties, UC3: View/Modify Properties, UC4: Add/Remove Units, UC5: View/Modify Units, UC10: Add Rental Agreements, UC11: View/Modify Rental Agreements
 - i. These requirements were partially completed in this iteration. The initial view designs required modifying and the overall set of views need more refining.

2.2 Appendix A

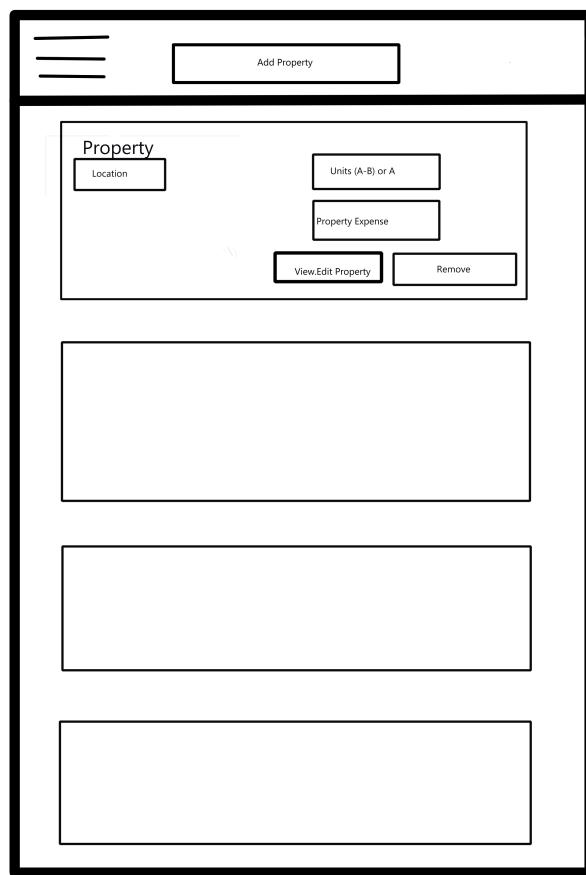


Figure 1: Property view

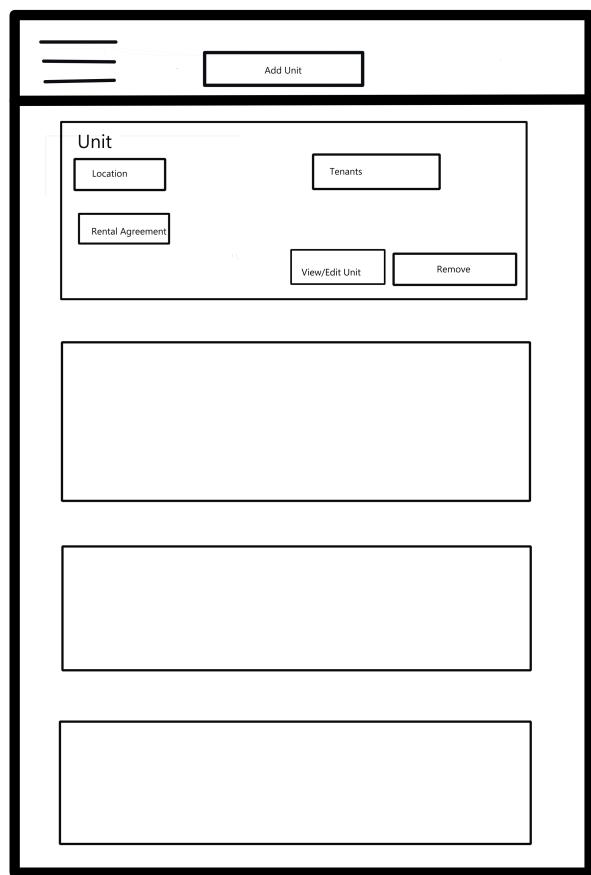


Figure 2: Unit View

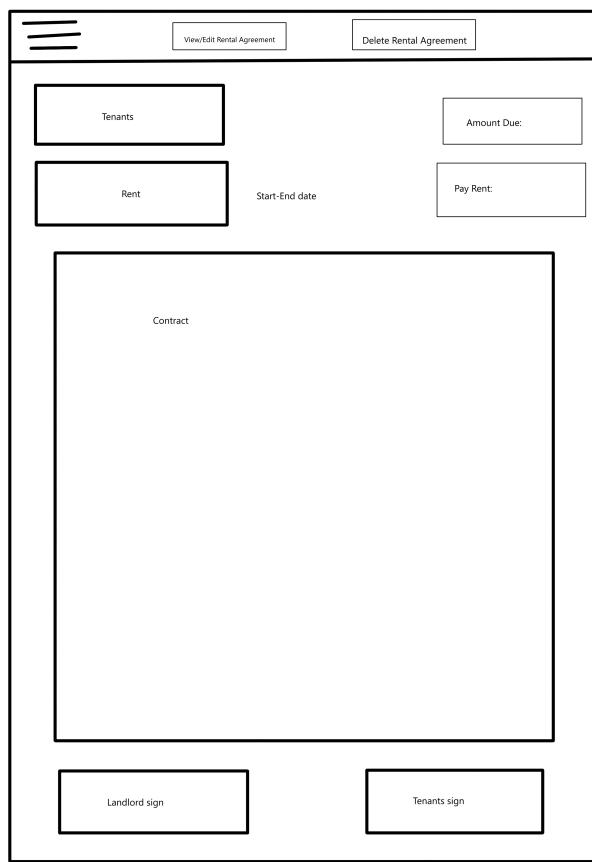


Figure 3: Rental Agreement View