Nicholas Arlinghaus
CS 5002: Senior Design II
Professor Fred Annexstein

Test Plan

Overall Test Plan

The primary testing strategy employed by this project is to create “dummy data” and ensure functionality of key features. While tools exist for creating dummy data at scale, the data will likely be created by hand through the software’s interface. This, in itself, is also a test of the data entry feature. Fake tenant names, addresses, contact info, balance info, and resource meter readings will all be inserted, and various reports will be created to break it all down.

Test Case Descriptions

1. Data Entry Test 1
    1. DE1
    2. Ensure tenant addition feature properly adds tenant data to database
    3. Dummy tenant data will be loaded via the “new tenant” feature in the “tenant info” menu. The information will then be examined in the database to ensure each piece of data is in the correct place.
    4. Tenant first and last name, employer, employer phone, employer name, move-in date, deposit amount, rent amount, weekly due day, payment frequency, next of kin name, next of kin relationship, next of kin phone, tenant cell phone, notes.
    5. Each piece of data will be inserted into the proper area of the database.
    6. Normal use case
    7. Blackbox
    8. Unit test
    9. **Results:** Pass. Tenant data added to proper table, each attribute added to proper column.
2. Data Entry Test 2
    1. DE2
    2. Ensure gas entry feature properly adds gas meter reading data to database
    3. Dummy gas meter readings will be loaded via the “add gas entry” feature on the “resources” page. The information will then be examined in the database to ensure each piece of data is in the correct place.
    4. Gas meter reading, date.
    5. Each piece of data will be inserted into the proper area of the database.
    6. Normal use case
    7. Blackbox
    8. Unit test
    9. **Results:** n/a, not implemented yet
3. Data Entry Test 3

    1. DE3
    2. Ensure water entry feature properly adds water meter reading data to database.
    3. Dummy water meter readings will be loaded via the “add water entry” feature on the “resources” page. The information will then be examined in the database to ensure each piece of data is in the correct place.
    4. Water meter reading, date.
    5. Each piece of data will be inserted into the proper area of the database.
    6. Normal use case
    7. Blackbox
    8. Unit test
    9. **Results:** n/a, not implemented yet
4. Navigation Test 1
    1. NAV1
    2. Ensure all interface buttons and navigation links work as intended.
    3. View-by-view, each button leading to a new view will be clicked to ensure that each one points to where it needs to go.
    4. Click input on each navigation button.
    5. Each button takes the user to the view it advertises.
    6. Normal use case
    7. Whitebox
    8. Integration test
    9. **Results:** Partial pass. Utility entry and report generation are not implemented yet however the features that do exist navigate correctly.
5. Reports Test 1
    1. YZF-R1
    2. Ensure each report request is handled properly and the necessary data is used to generate it.
    3. After adding dummy data, each report option will be selected and given a timeframe to generate a report. The report will be inspected and compared against a manually-created report to ensure the information is accurate.
    4. Dummy property (rent, resource) data, click input on each report option.
    5. Each report option generates accurate reports based on the data its asked to use.
    6. Normal use case
    7. Whitebox
    8. Integration test
    9. **Results:** n/a, not implemented yet
6. Reports Test 2
    1. R2
    2. Ensure print options correctly send generated report to printer.

3. The reports generated in YZF-R1 will be sent to the printer via the “print” icon. The two “quick print” options on the main page will also be clicked to ensure printer functionality.
4. Mouse click on the print icon after generating a report, or a mouse click on one of the two “quick print” options on the main page.
5. Each generated report sent to the printer will print as expected.
6. Normal use case
7. Whitebox
8. Unit test
9. **Results:** n/a, not implemented yet

7. Currently Due Test 1
    1. CD1
    2. Ensure that the “currently due” functionality works as expected.
    3. After dummy data is loaded, a simulation “rent collection” will take place in which tenants marked as “due” will “pay” their rent, both with or without a balance. The “Paid?” button will be clicked and the “Save” button later to ensure paid data gets sent to the database.
    4. For tenants without a balance, a mouse click on the “Paid?” button next to their name as well as the “Save” button in the corner. For tenants with a balance, a mouse click on the “Paid?” button next to their name, an input of how much they “paid”, and a mouse click on the “Save” button in the corner.
    5. After saving the changes made in the sheet, each payment will be recorded in the database with the proper amount.
    6. Normal use case
    7. Blackbox
    8. Integration test
    9. **Results:** Pass. Each payment amount is properly sent to the database upon pressing “save”.
8. Currently Due Test 2
    1. CD2
    2. Ensure the “undo” functionality works as intended.
    3. A payment will be recorded as described in CD1, however without clicking the “Save” button. The last payment will be followed by another click on the “Paid” button, thus leaving the tenant’s information in the table and not submitting their payment upon pressing "Save”.
    4. Mouse click on the “Paid” button as well as the “save” button.
    5. The payment, once undone, will not appear in the database after clicking the “save” button.
    6. Normal use case.
    7. Blackbox
    8. Integration test
    9. **Results:** Pass. A payment made and then undone was not present in the database after pressing “Save”.

9. Tenant Info Test 1
    1. TI1
    2. Ensure the “view” tenant information buttons work as intended on the “tenant info” page.
    3. Each view – “Current Tenants”, “Past Tenants”, and “Everyone” will be selected and each view inspected/compared to database entries to ensure the correct tenants are being shown in the view.
    4. Mouse click on one of the three options, mouse click on all of the dropdown options for specific building or all.
    5. The tenants shown will be consistent with the parameters defined (current vs past, building, etc)
    6. Normal use case
    7. Blackbox
    8. Integration test
    9. **Results:** Pass. Each button successfully navigates to the section it describes and the data displays correctly.
10. Insights Test 1
    1. IT1
    2. Ensure the insights shown on each screen are accurate
    3. The “Insights” on the left side of some views display relevant information generated from the data in the database. This test ensures this information is accurate.
    4. Data in database.
    5. The insights it generates are all numerically accurate.
    6. Normal use case
    7. Blackbox
    8. Unit test
    9. **Results:** n/a, not implemented yet

Test Case Matrix

Test	Normal/Abnormal	Black/Whitebox	Integration/Unit
DE1	Normal	Blackbox	Unit
DE2	Normal	Blackbox	Unit
DE3	Normal	Blackbox	Unit
NAV1	Normal	Whitebox	Integration
YZF-R1	Normal	Whitebox	Integration
R2	Normal	Whitebox	Unit
CD1	Normal	Blackbox	Integration
CD2	Normal	Blackbox	Integration
TI1	Normal	Blackbox	Integration
IT1	Normal	Blackbox	Unit