THE MYWICK PROPERTY MANAGEMENT SYSTEM

Team Scandians

Naresh Choudhary
Ashwin Sunil Bhide
Shikhar Chandra
Soma Sunder Gutala
Shuang Li
Shruthi Shetty
Pallavi Tiwari

Date: 12/04/2018

ISTM624- Section 602

Table of Contents

Quality Control Document

Sr. No.	Document Title	Version	Last Modified	Document Owner	Reviewed By	Quality Check	Initials
1	Executive Summary	1.4	11/05/2018	Ashwin Bhide	Naresh Choudhary	Done	AB
2.	Class Diagram	1.8	11/30/2018	Ashwin Bhide	Naresh Choudhary, Shuang Li, Shikhar Chandra	Done	AB
3.	Deployment Diagram	1.4	11/29/2018	Shikhar Chandra	Naresh Choudhary, Ashwin Bhide	Done	SC
4.	Data Flow Diagram	1.6	12/03/2018	Naresh Choudhary	Pallavi Tiwari, Shruthi	Done	NC
5.	Sequence Diagram	1.9	12/01/2018	Pallavi Tiwari, Shuang LI	Naresh Choudhary	Done	PT, SL
6.	Testing Plan	1.3	12/03/2018	Soma Sunder	Shruthi Shetty	Done	SS
7.	Security Plan	1.2	12/01/2018	Shruthi Shetty	Soma Sunder	Done	SS
8.	Scrum Essay	1.2	11/30/2018	Shruthi Shetty	Ashwin Bhide	Done	SS
9.	Waterfall Essay	1.2	11/25/2018	Shuang Li	Pallavi Tiwari	Done	SL
10.	Agile Essay	1.2	11/27/2018	Shikhar Chandra	Shuang Li	Done	SC
11.	User Story	1.6	12/01/2018	Shuang Li, Pallavi Tiwari	Naresh Choudhary	Done	SL,PT

Executive summary

Ms.Mywick, a real estate owner, owns several properties; each property containing a fixed number of units. She is an absentee property manager – meaning she does not reside on property and manages the rental system remotely. Currently, Quick Books software is being used to track expenses and rents are being tracked manually. There are several software packages available in the market to track rents which offer overly complex features that are not required and hamper the ease of use of the system. We are proposing a system that offers front-end data entry and back-end database system with report generation capabilities. In addition, our system will integrate with Quick Books by exporting transaction data to facilitate generation of income reports. Also, owing to the familiarity of the owner with Windows applications, we are following similar GUI style.

Sometimes part of tenants' rent is paid in the form of federal subsidies or other sources. The rent needs to be paid on a current monthly basis. The property owner needs to track income from rent on the basis of tenant, unit, property as well as in aggregate. A software system which is easy to use, integrates well with the Quick Books software and capable of recording received and unpaid rents is the requirement of the owner. In addition, the software must be able to receive payments receipts from multiple sources and process transactions in batch mode.

Based on the conversation with the property owner, handling of utility bills is not part of the system. The software system does not require a secure-login functionality and tracking of property expenses is not included in this software. Payment receipts do not need to be generated. The personal information related to each tenant which is captured by the system does not need to be elaborate. Tracking of deposit refunds is an optional requirement and may be included in the future scope.

Our solution to this problem is a software system to automate the process of tracking rents. The front-end of the system will include a user interface to capture property, unit and tenant details. A switchboard menu will be provided to enter payments, add units and update property details. Also, a brief description about each unit can be input to categorize the property type. The system will be able to accept the itemized subsidy list provided by HAP (Housing Assistance Program) and auto-populate the amount of subsidy associated with each tenant. The user interface accepting rental checks is robust and will include check-boxes, radio-buttons and drop-down lists to ensure consistency in storage and retrieval of data. The back-end storage system will consist data which changes frequently – source of payments and subsidy amount while some data will remain static – addresses of property units. The payment history of each tenant will be persisted into the database along with the tenant information. Report generation functionality based on rent paid and rent owed will be provided. Additionally, reports displaying the monthly payment history, rent balance, total income by unit, lease deadlines as well as tenant defaulter list will be provided. As a result, the owner will have the option of writing off unpaid rent for a tenant based on previous track record.

In conclusion, the software system will efficiently track the rent payments received versus the payments due thereby reducing the manual effort and automating the rent tracking process. An easy to use front-end interface with rich back-end services is the highlight of our proposed system. Moreover, the critical insights provided by the rental reports will assist the owner in effective property management.

User Story

1. As an employee, I want to add and edit basic tenant information, like name, phone number and deposit amount paid, so that I can track their information and contact them if necessary.
   1.1. Given the employee wants to enter the information for a tenant, user should be able to enter first name, last name, phone number and deposit amount paid in the tenant information form.
   1.2. Given that the employee has entered correct details and the fields have been validated, the user should be able to save this information by clicking on “Save” button.
   1.3. Given the employee has entered data in the fields, there should be data validations in place to facilitate storing clean data.
   1.4. Given that the employee is opening the information of an existing tenant, user should be able to edit the details and save them.
2. As an employee, I want to enter the actual property and unit information and make changes to them afterward, so that I can track all the information about the property.
   2.1. Given the employee wants to save the property details, user should be able to enter the details related to a property such as tenant, unit, type of property, unit description, and price and save them to the database.
   2.2. Given the property details are saved, user can edit the details and save them again.
   2.3. Given there is data available, user can track it by property, by unit, by tenants, and by payment.
3. As an employee, I want to enter and edit lease information for each tenant, like the property type, unit number, rent amount, so that I can track all the rent information and know the payment due for each month.

3.1. Given the employee is logged in, the user can enter and update the lease information.

4. As an employee, I want to record checks from tenants at various times over the course of a month in batch mode, like rent payment time, tenant name and amount, so that I can know the income of each month.

4.1. Given the employee has received a batch of checks one time, when the user wants to record all of the information, he/she can record them in a batch mode.

4.2. Given the employee has received a batch of checks one time, when the user wants to enter the amount of each check from a tenant, the expected rent amount is displayed on the form.

4.3. Given the employee keeps receiving checks in a month, user can run this batch process multiple times in a month successfully.

4.4. Given the checks are received, employee can view and update the rent amount received from a tenant in a form with a grid-like structure.

5. As an employee, I want to record information of payments from HAP, like tenant name, reference and amount, so that I can know the income of each month.

5.1. Given the employee has received the HAP check, when the user selects “enter HAP payment for this month”, then the system should cycle through the tenants that have subsidy payments due.

5.2. Given the employee has received the HAP check, when the user selects “enter HAP payment for this month”, then the system should pre-populate subsidy amount fields where possible.

5.3. Given the employee has received the HAP check, when the user selects “enter HAP payment for this month”, then the system should show the expected subsidy amount.

5.4. Given the employee has received the HAP check, when the user selects “enter HAP payment for this month”, then the system should allow the user to change the expected amount, memorize the amount and re-calculate the amount expected from the tenant.

6. As an employee, I want to know the rents owed for each tenant and handle recurring amounts based on a one-year lease, so that I will know how much rent a tenant should pay for each month.

6.1. Given the employee is logged in, the user can view the rents owed for each tenant yearly.

6.2. Given the employee is logged in, the user can view the amount the tenant needs to pay per month.

7. As an employee I want to adjust owed payment of each tenant monthly, so that I can get the accurate and updated information for each month.

7.1. Given the employee is logged in and is viewing the rent information for a tenant, the user is able to change the rent owed and make monthly adjustments.

7.2. Given the employee has made changes, these changes to the monthly adjustments should get updated in the system successfully.

8. As an employee, I want to know the multiple sources of payments of the tenant, so that I can identify the total amount due from the tenant.

8.1. Given the employee is logged in, the system can provide for multiple sources of payments such as federal subsidy, multiple tenants each paying a portion, parents paying part of the rent, HAP subsidy.

8.2. Given the system handles multiple resources, the system should be able to update the rent owed amount whenever there is any change in the income of the tenant or the HAP subsidy amount.

9. As an employee, I want to adjust the amount of rent for each tenant, so that I can write off some amount of rent for some particular tenants.

    9.1. Given the employee wants to reduce some amount of the rent, when he/she selects “edit rent amount”, then a comment field will be displayed for descriptions of reasons.

10. As an employee, I want to track outstanding amounts due and overdue rent from previous months on the first day of the month, including tenant information, so that I can send rent notifications to the tenants to help them pay on time.

    10.1. Given the employee logs in, when he/she selects “generate amount due report (last month)”, then the due amount for this month and previous periods are displayed.

    10.2. Given the employee logs in, when he/she selects “generate amount due report (last month)”, then she can choose the reporting period, like last month, or last 60 or 90 days.

    10.3. Given the employee logs in, when he/she selects “generate amount due report (last month)”, then the current monthly payment history by unit showing rent due, payments made and balance due are displayed.

11. As an employee, I want to generate total income reports by unit by month and by the tenant by month, so that I can know all the income information.

    11.1. Given the employee logs in, when he/she selects “generate income report by tenant”, then the income reports by month and by the tenant (current tenants and past tenants) is displayed.

12. As an employee, I want to export data into QuickBooks, so that I can do calculations and manage financial information.

    12.1. Given the data is available for export, QuickBooks set up should be done on the user’s machine.

12.2. Given that QuickBook is installed, the user can export data using excel/.csv files to the Quick Book software.

13. As an employee, I want to know how many properties are available, so that I can adjust my business plan to increase profit.

13.1. Given there is information about the property in the database, the user can query this and create a report of the list of current vacancies.

13.2. Given the system generates reports, the time taken to load these reports should be less than 30 seconds.

14. As an employee, I want to log in and log out to the system, so that I can perform operations in the system.

14.1. Given the employee wants to use the system, when the user selects “Log in”, then the login interface is displayed to let the user enter the account name and password.

14.2. Given the employee doesn’t want to use the system when the user selects “Log out”, then the user will log out safely and the information will be saved safely.

14.3. Given more than one employee wants to use the system at the same time, when the second user logs in, all the users can have full capabilities.

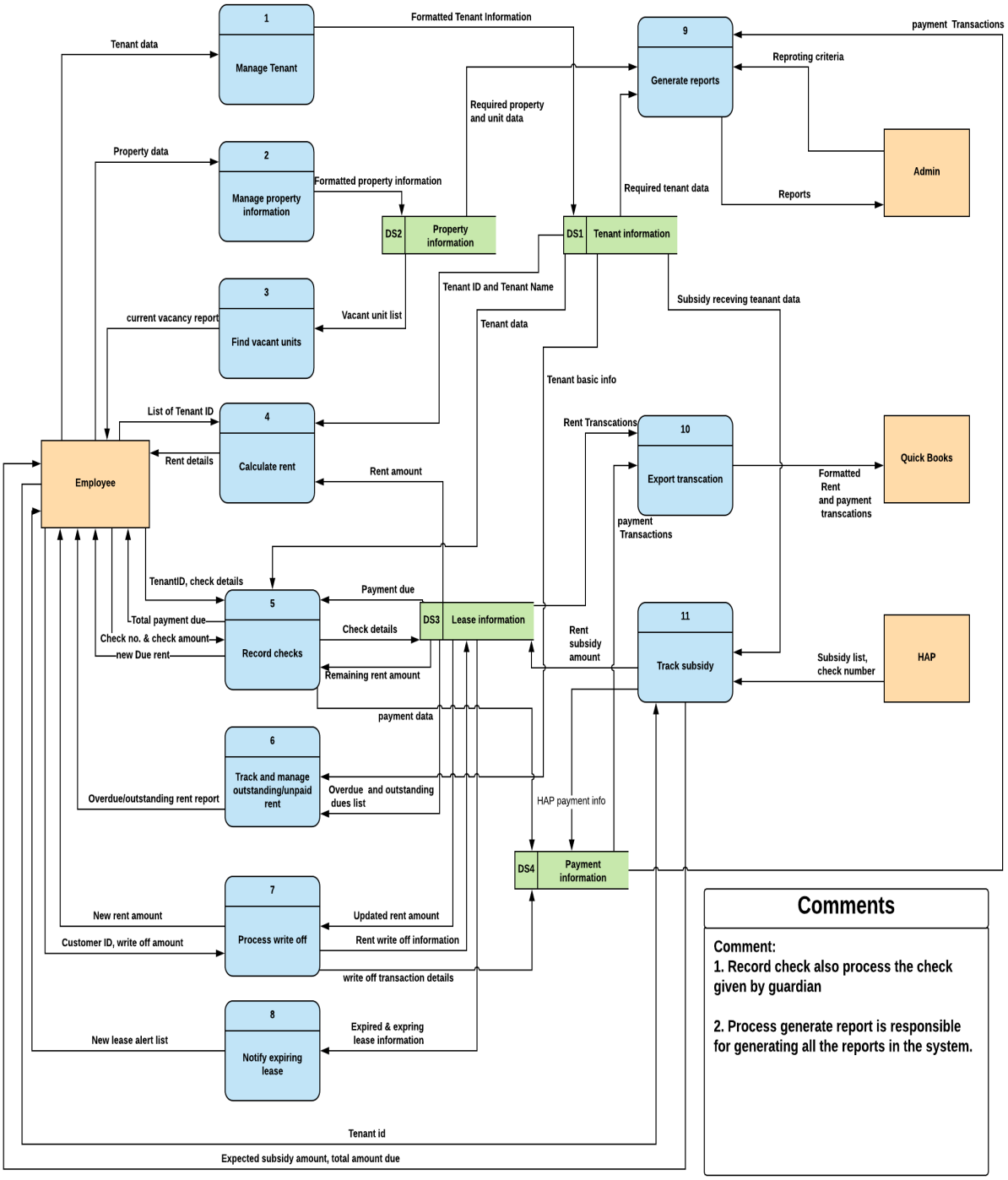
14.4. Given the employee is logged in for more than 20 minutes, the system will log him/her out due to session timeout.

Overall Architecture Analysis

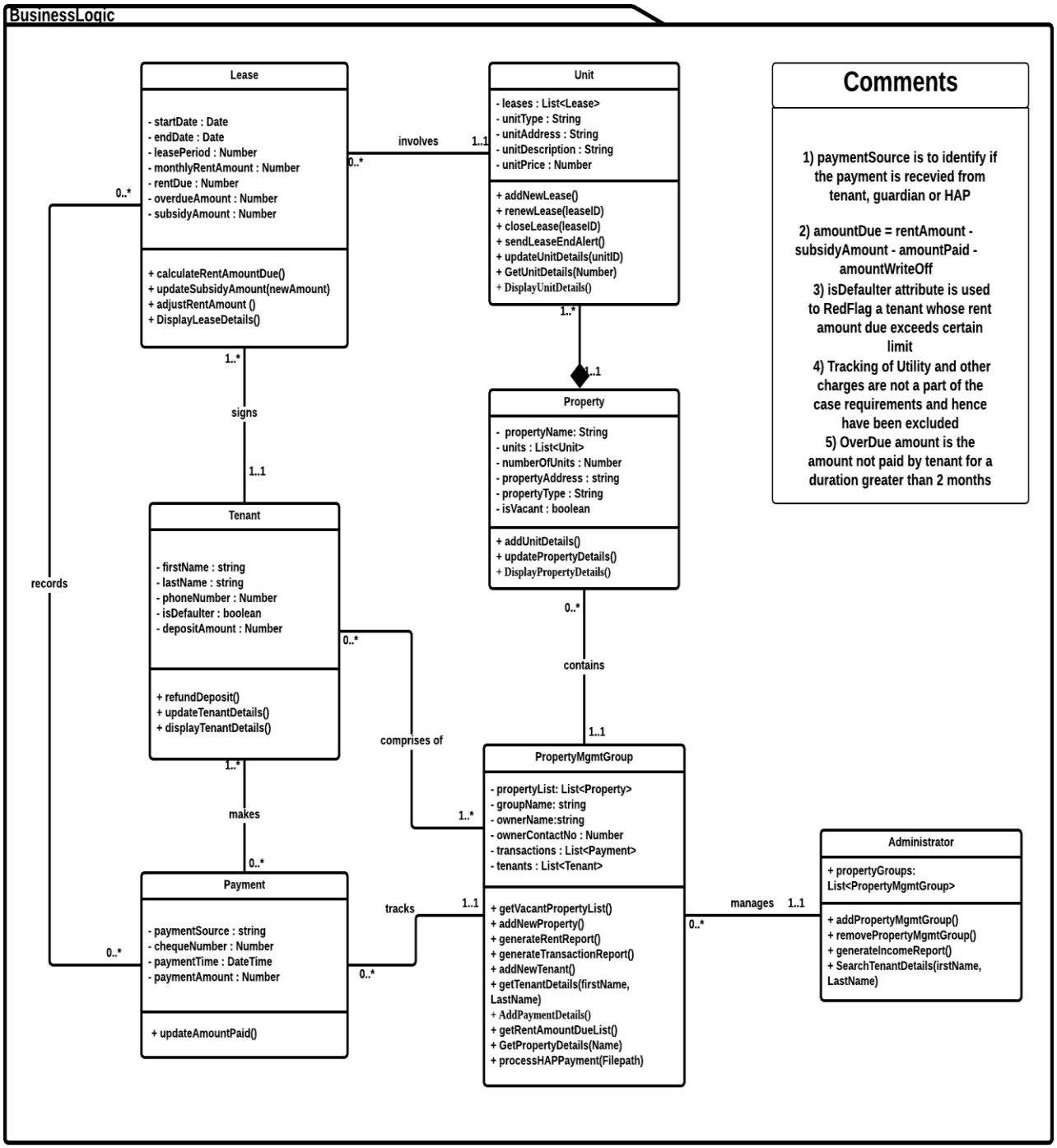
Property Management System is a robust yet simple product which allows property owners to perform daily business task in an easy and efficiently manner. It can handle up to 10,000 properties and 10,000 units in each property. Primary capabilities of the system are to generate various revenue reports, store checks in batch mode and send notifications related to lease and payments. Another key feature of the system is to process partial rent payments which will allow user to record any amount of payment; system will automatically calculate the rent due. It can be easily integrated with external systems such as Quick books and HAP. As quick books tracks the expenses, currently system does not record any expense related transactions.

Four major external entities in our system are Admin, employee, quick books and HAP. Admin are the owner of properties managing the business. Depending on the case, a user can have both admin and employee level access. Another entity is quick book which will only take input from our system. Since quick book provide the feature of tracking expense and generating standard reports, we have only include the feature to store rent transactions and payment associated with them. According to the case, output from quick book is external to the system. Lastly, HAP is a government organization which sends a single check with attached file containing name of tenant receiving subsidy and subsidy amount. To hold this information for auditing and tracking purpose, process ‘track subsidy’ creates records in payment database.

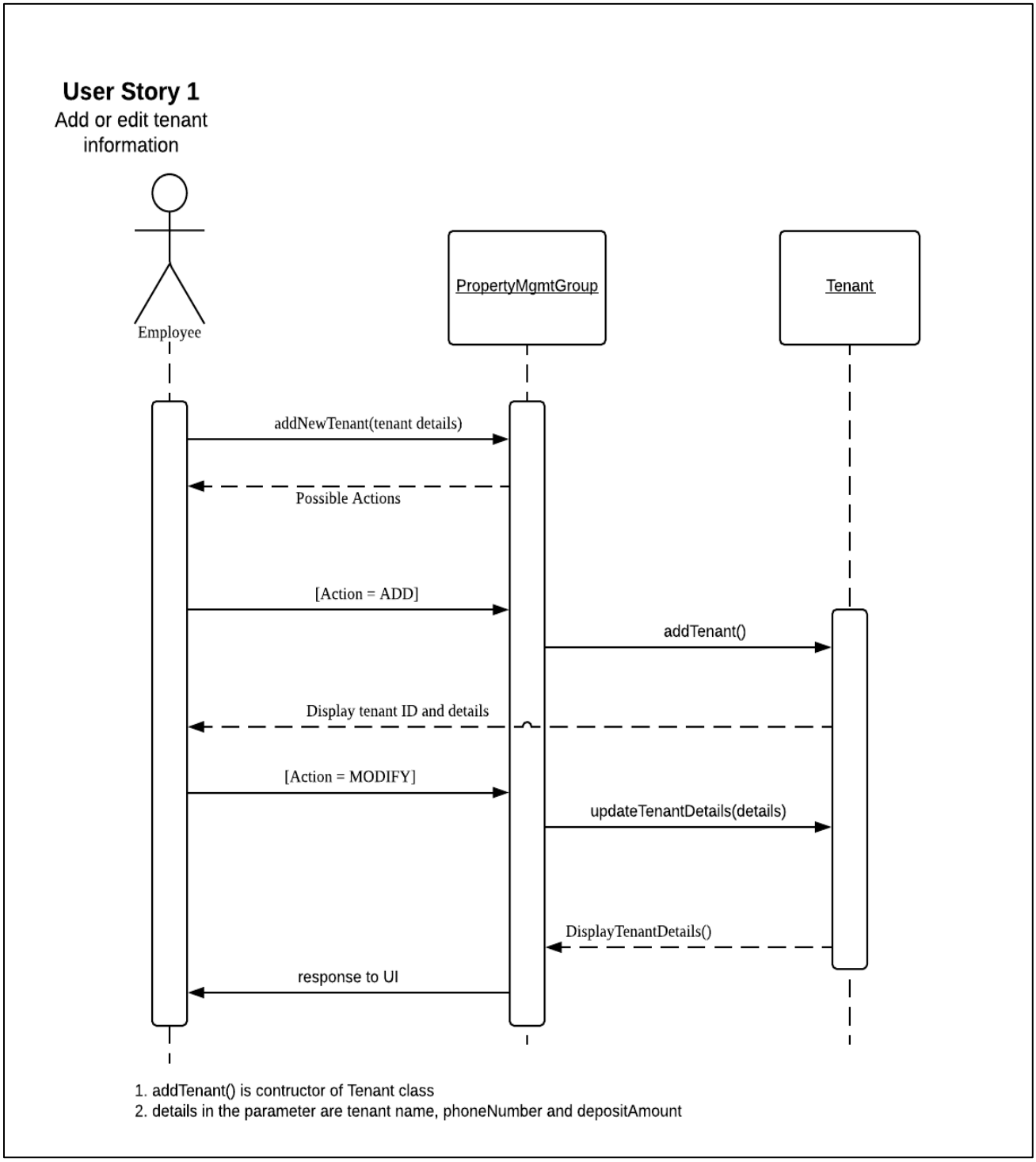
To provide smooth and faster experience, development is done in n tier architecture. It has three app servers to support the application and balance the load. Two database are used store the data which are synched by a database server. Since data is of top concern in our business, another backup database is used which archives data on daily basis.

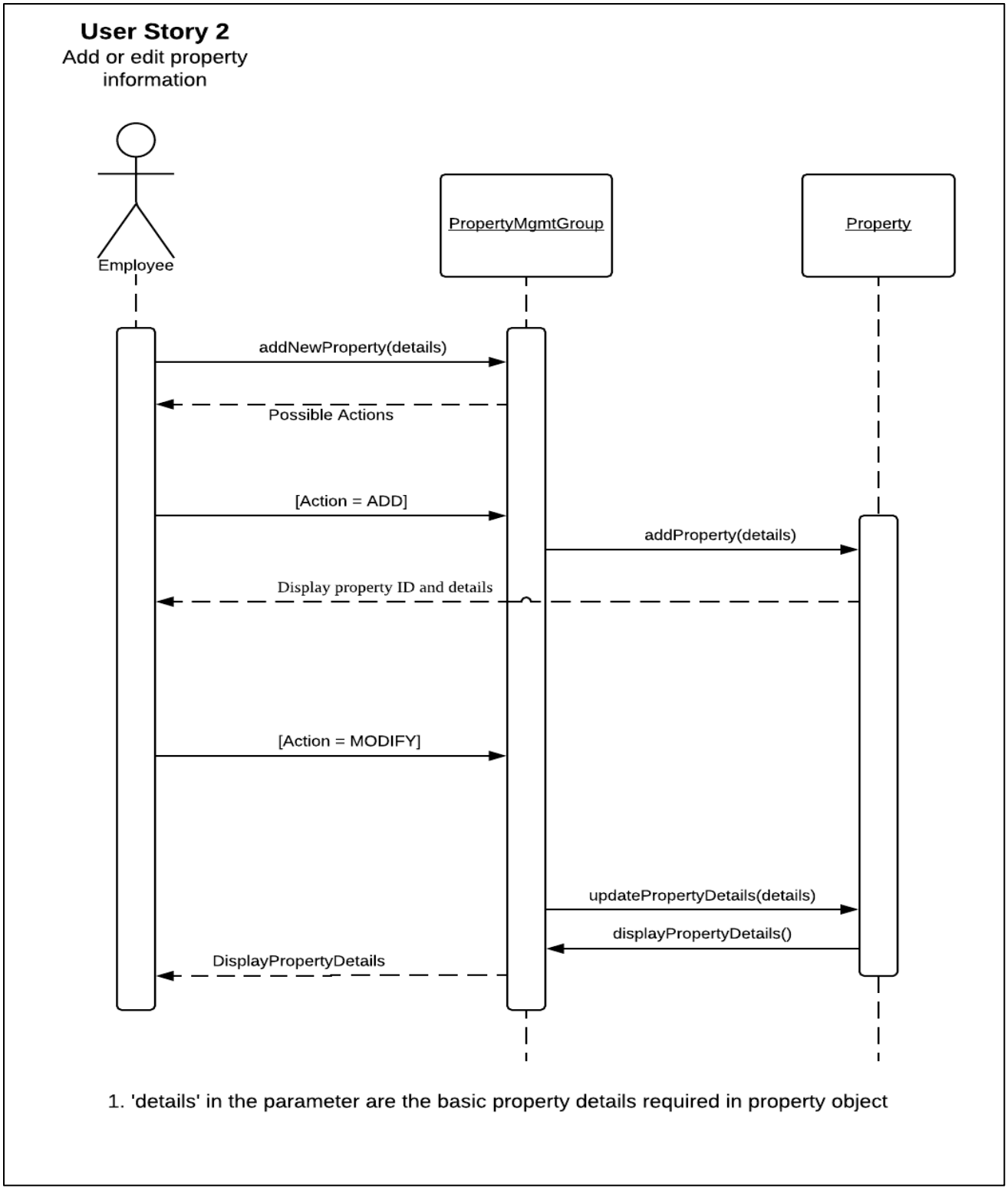

1
Manage Tenant
Tenant data
Formatted Tenant Information
9
Generate reports
payment Transactions
Reproting criteria
Admin
Reports
Required property and unit data
Required tenant data
2
Manage property information
Property data
Formatted property information
DS2 Property information
DS1 Tenant information
3
Find vacant units
current vacancy report
Vacant unit list
Tenant ID and Tenant Name
Subsidy receving teanant data
Tenant data
Tenant basic info
4
Calculate rent
List of Tenant ID
Rent details
Rent amount
Rent Transcations
10
Export transcation
Quick Books
Formatted Rent and payment transcations
payment Transactions
Employee
TenantID, check details
Payment due
5
Record checks
Total payment due
Check no. & check amount
new Due rent
Check details
Remaining rent amount
DS3 Lease information
Rent subsidy amount
11
Track subsidy
Subsidy list, check number
HAP
payment data
6
Track and manage outstanding/unpaid rent
Overdue/outstanding rent report
Overdue and outstanding dues list
HAP payment info
DS4 Payment information
7
Process write off
New rent amount
Customer ID, write off amount
Updated rent amount
Rent write off information
write off transaction details
8
Notify expiring lease
New lease alert list
Expired & expring lease information
Tenant id
Expected subsidy amount, total amount due
Comments
Comment:
1. Record check also process the check given by guardian
2. Process generate report is responsible for generating all the reports in the system.

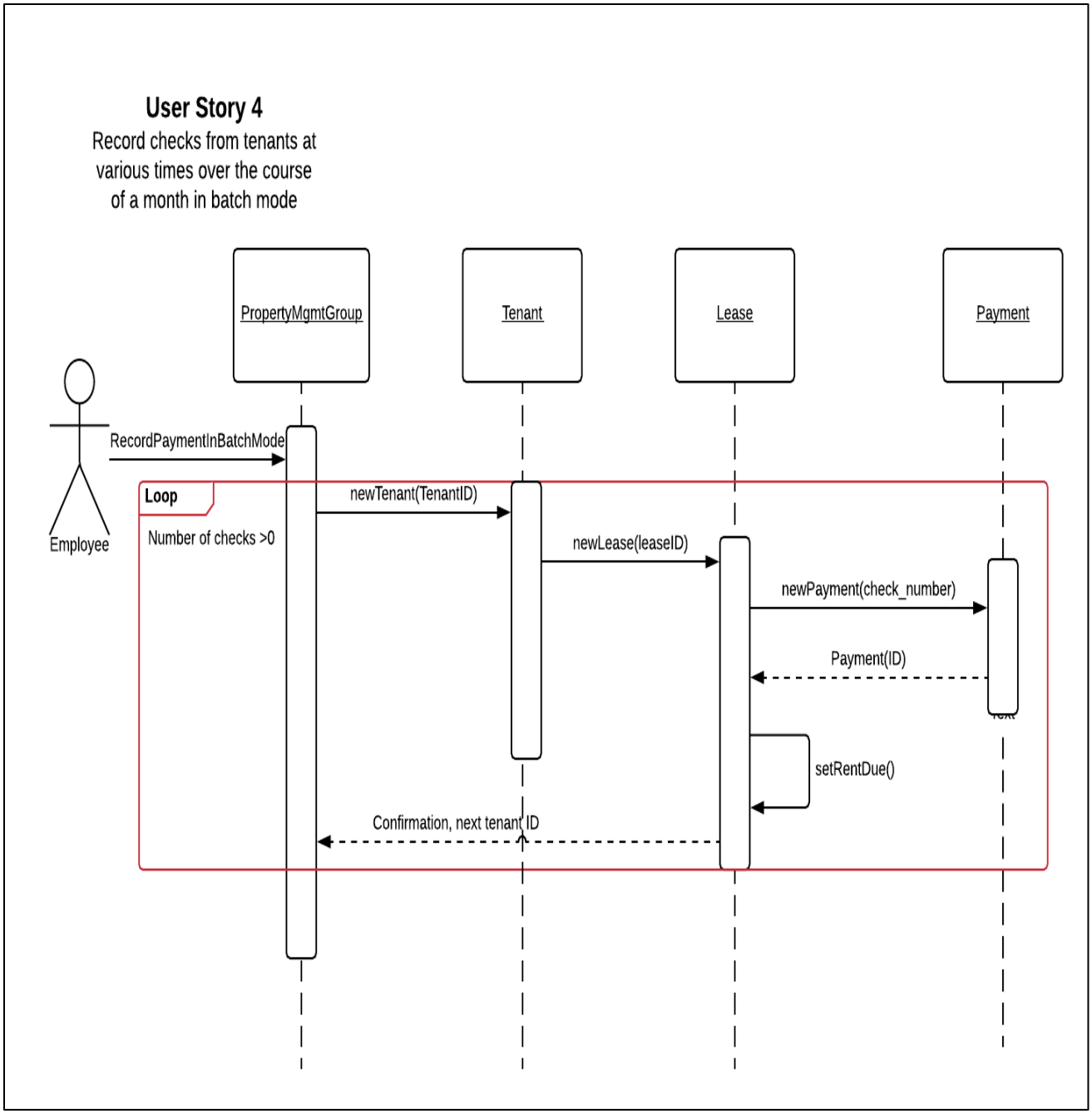
UML Class Diagram

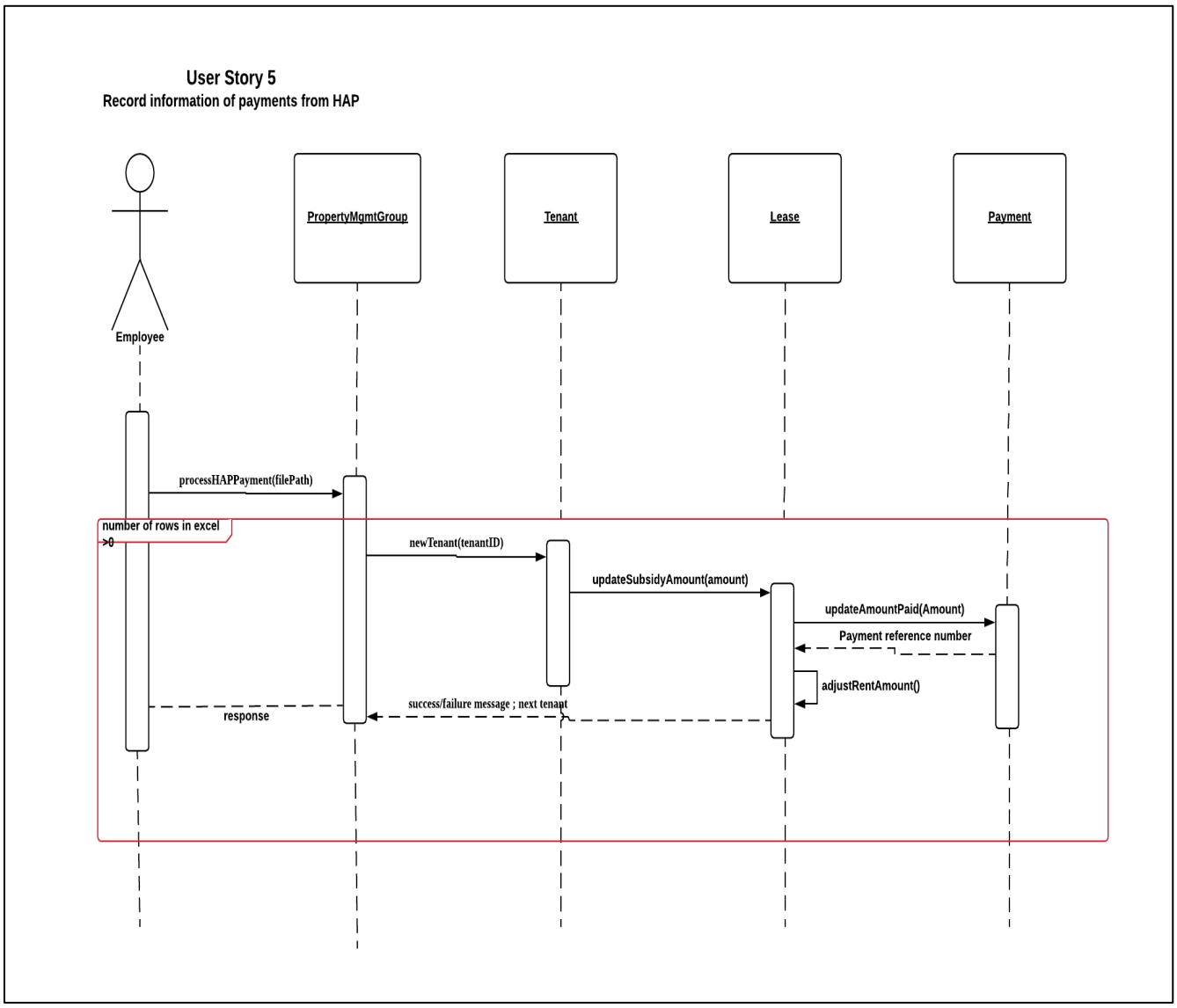


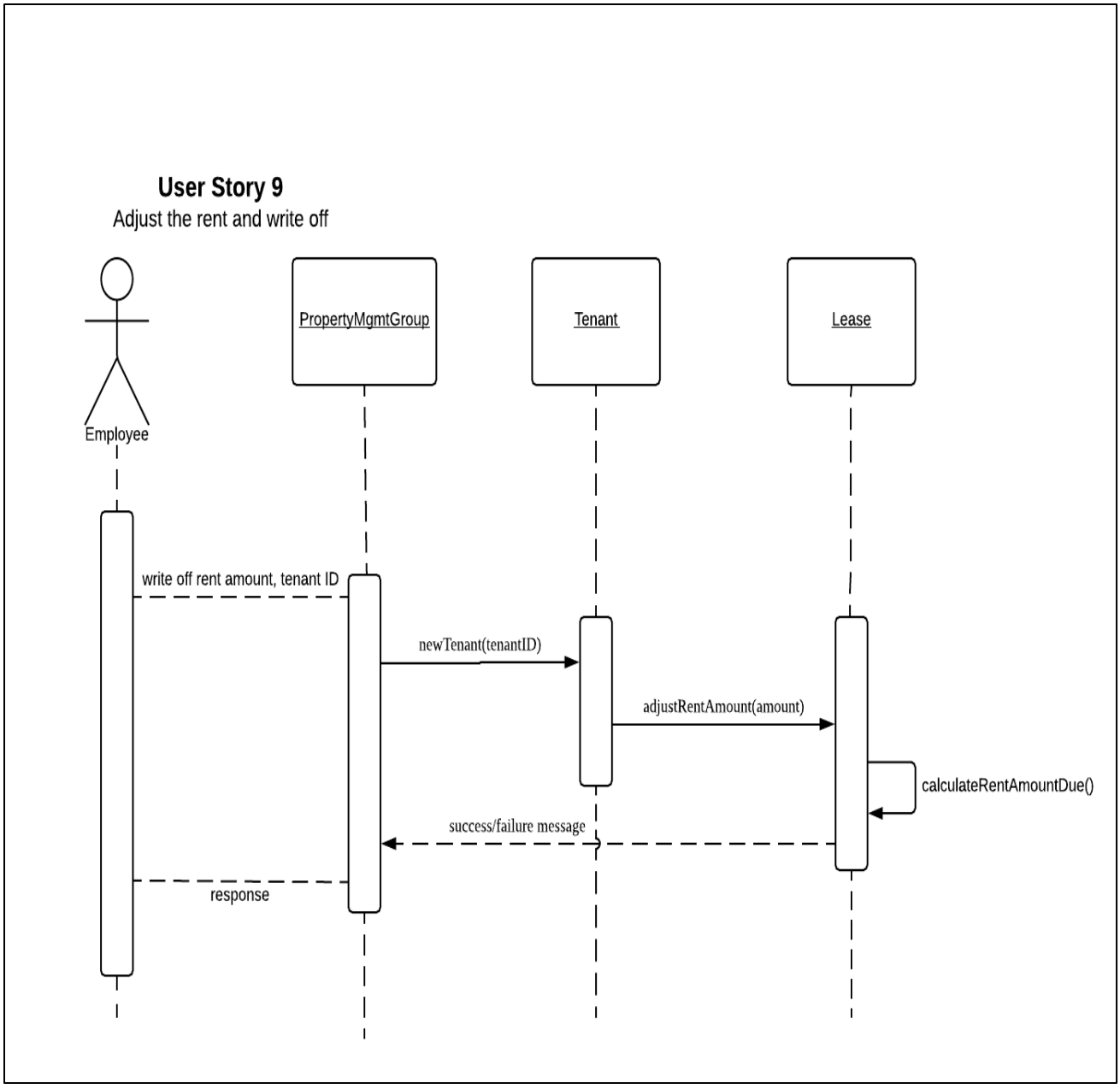
UML Sequence Diagrams

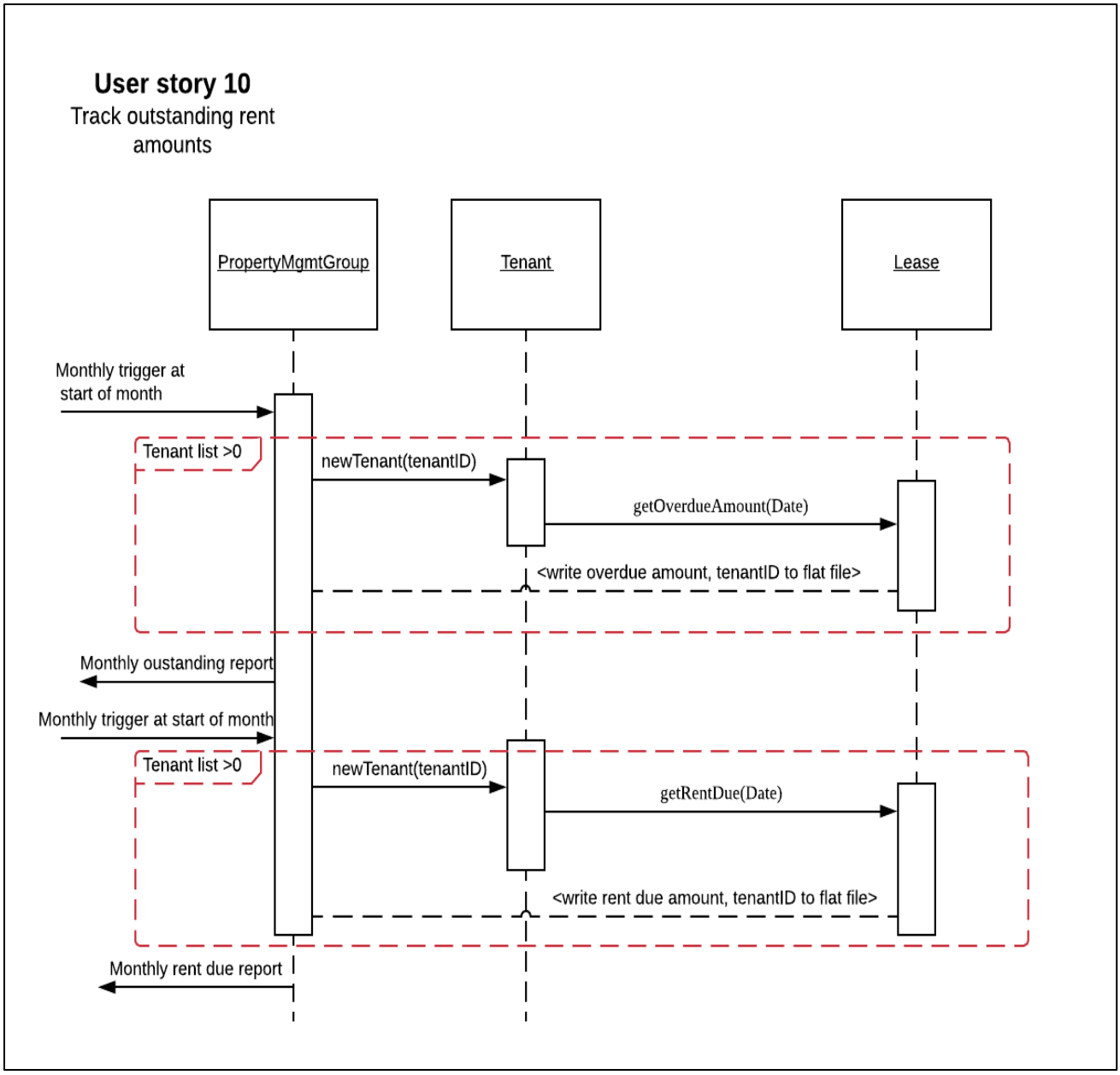


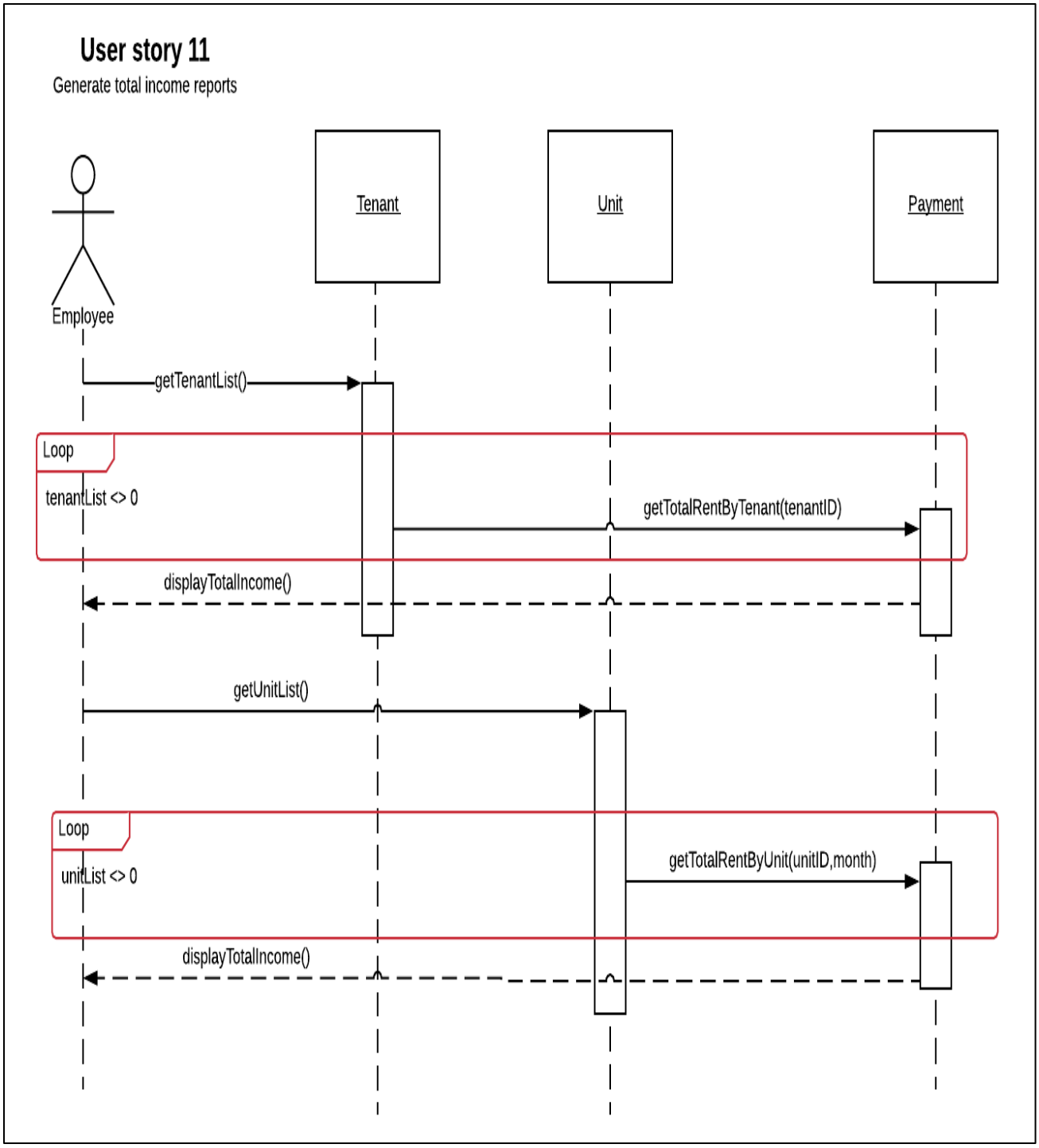

User Story 2
Add or edit property information
Employee
PropertyMgmtGroup
Property
addNewProperty(details)
Possible Actions
[Action = ADD]
addProperty(details)
Display property ID and details
[Action = MODIFY]
updatePropertyDetails(details)
displayPropertyDetails()
DisplayPropertyDetails
1. 'details' in the parameter are the basic property details required in property object

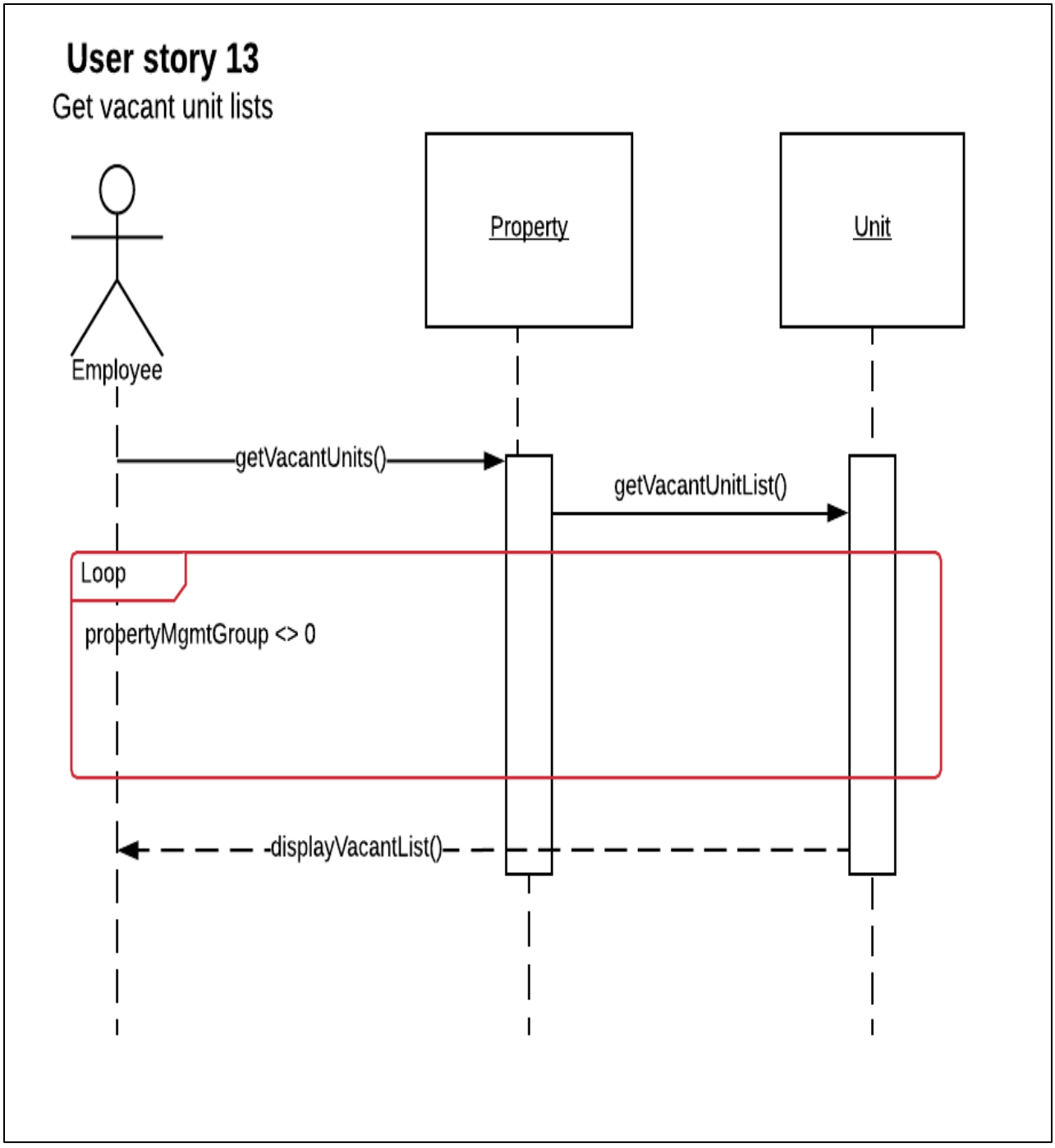

User Story 4
Record checks from tenants at various times over the course of a month in batch mode
PropertyMgmtGroup
Tenant
Lease
Payment
Employee
RecordPaymentInBatchMode
Loop
Number of checks >0
newTenant(TenantID)
newLease(leaseID)
newPayment(check_number)
Payment(ID)
setRentDue()
Confirmation, next tenant ID


User Story 5
Record information of payments from HAP
Employee
PropertyMgmtGroup
Tenant
Lease
Payment
processHAPPayment(filePath)
number of rows in excel
>0
newTenant(tenantID)
updateSubsidyAmount(amount)
updateAmountPaid(Amount)
Payment reference number
adjustRentAmount()
success/failure message ; next tenant
response


User Story 9
Adjust the rent and write off
Employee
PropertyMgmtGroup
Tenant
Lease
write off rent amount, tenant ID
newTenant(tenantID)
adjustRentAmount(amount)
calculateRentAmountDue()
success/failure message
response

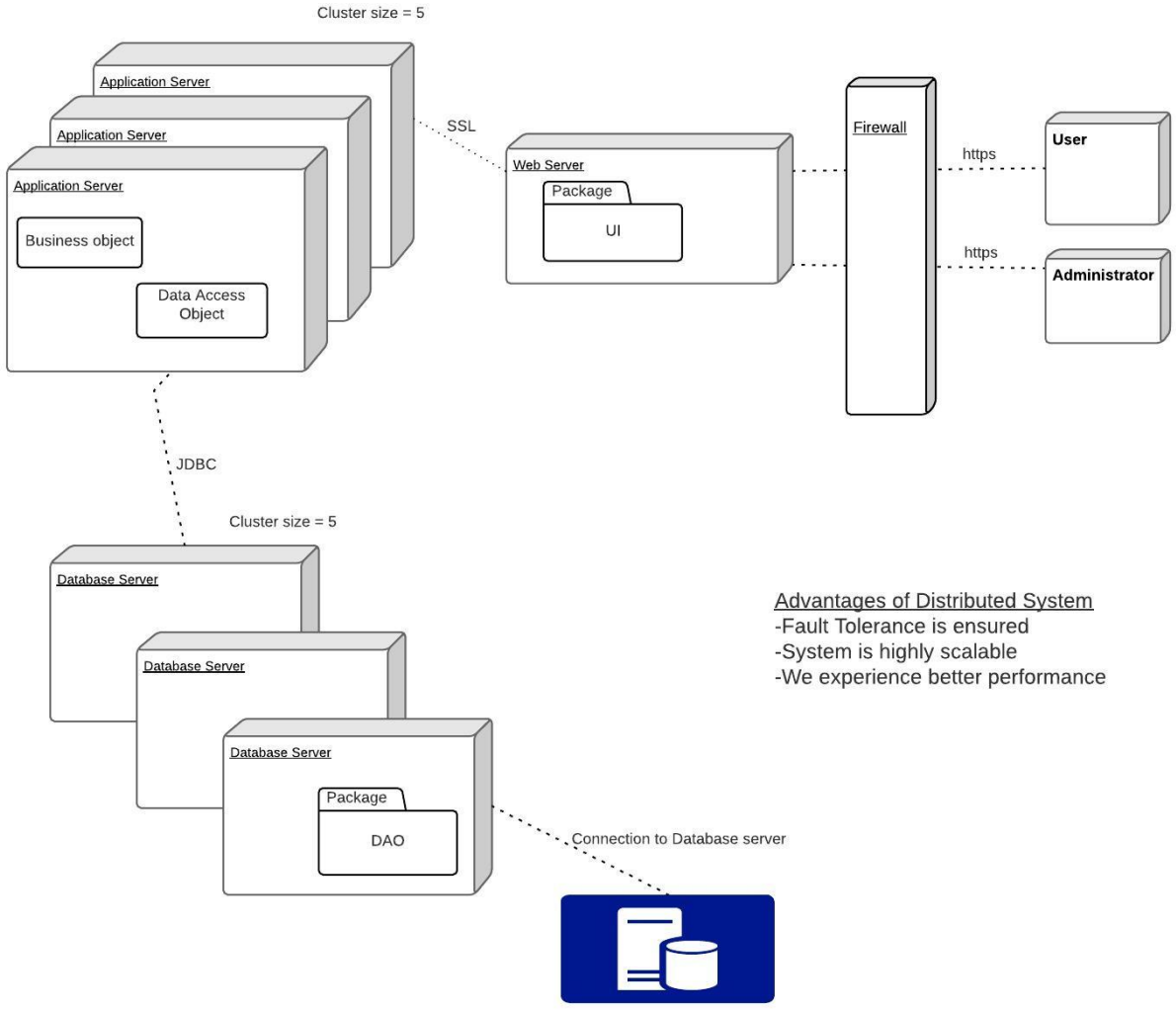

User story 10
Track outstanding rent amounts
PropertyMgmtGroup
Tenant
Lease
Monthly trigger at start of month
Tenant list >0
newTenant(tenantID)
getOverdueAmount(Date)
<write overdue amount, tenantID to flat file>
Monthly oustanding report
Monthly trigger at start of month
Tenant list >0
newTenant(tenantID)
getRentDue(Date)
<write rent due amount, tenantID to flat file>
Monthly rent due report


User story 11
Generate total income reports
Employee
Tenant
Unit
Payment
getTenantList()
Loop
tenantList <> 0
getTotalRentByTenant(tenantID)
displayTotalIncome()
getUnitList()
Loop
unitList <> 0
getTotalRentByUnit(unitID,month)
displayTotalIncome()


User story 13
Get vacant unit lists
Employee
Property
Unit
getVacantUnits()
getVacantUnitList()
Loop
propertyMgmtGroup <> 0
displayVacantList()

UML Deployment Diagram

We have used an N-tier architecture in the form of a distributed system. The main advantage of using a distributed system is fault tolerance, scalability and better performance. It will ensure that the application is accessible on a server if one of the other servers go down.



Testing Plan

1. Introduction

Property Management offices require a system that offers front-end data entry and back-end database system with report generation capabilities. This property management system is a software where property details such as available house details, schedules, address, and others are been setup by an administrator[3]. This test approach document describes the appropriate strategies, process, workflows and methodologies used to plan, organize, execute and manage testing of the property management system [1].

The document mainly targets to ensure that the property management system meets all of its technical, functional and business requirements.

1.1. Scope

The document mainly targets to ensure that the property management system meets all of its technical, functional and business requirements [2]

1.1.1. In Scope

This *Test Plan* defines the unit, integration, system, regression, and Client Acceptance testing approach. The test scope includes the following functions to be tested [3]:

- Returning results concerning type of property, price details and other available property information
- Entering or updating the details regarding the primary, contact and other information of the tenants

- Accepting payments from multiple sources and sorting it accordingly
- Generating weekly or monthly reports
- Proper functioning of search and filter logics
- Performance of the system as a whole
- Easy UI
- Secure access to the system

1.1.2. Out of Scope

The following are considered out of scope for this *Test Plan* and testing scope:

- Functional requirement testing of software outside the system
- Disaster recovery and Business Continuity Plan.

1.2. Assumptions for Test Execution

Below are some minimum assumptions to be considered this project are as follows.

- For User Acceptance testing, the Developer team should complete unit, system and integration testing
- User Acceptance testing will be conducted by End-users
- Test scripts are developed and approved.
- Major dependencies should be reported immediately after the testing kickoff meeting.

2. Test Methodology

The document mainly targets to ensure that the property management system meets all of its technical, functional and business requirements.

2.1. Approach

2.1.1. Usability Testing

Usability testing is performed so as to ensure that the features of the system developed are satisfactory to the customer.

2.1.2. Unit Testing

Unit Testing is performed wherein each unit of the system developed is tested. Validation of each software component is performed.

2.1.3. Performance Testing

It is performed to determine the responsiveness, speed and stability of the system being developed [4]. Different types of performance testing that are performed are:

- Load Testing
- Endurance Testing
- Scalability Testing
- Stress Testing

2.1.4. Regression Testing

Regression Testing is performed to ensure that the new changes made to the system will not break the existing code or the functionalities. This is performed before the deployment of the new code into the system.

2.1.5. Security Testing

This testing intends to uncover vulnerabilities of the designed system and determine that its data and resources are protected from possible intruders [5].

2.1.6. Acceptance Testing

Acceptance Testing is performed to verify whether Product meets customer requirements for acceptability [6]. In this, all the user stories are checked whether they meet the acceptance criteria or not. Depending on the feedback, it is ensured that the system can move into production with no issues.

2.2. Suspension Criteria and Resumption Requirements

Testing is suspended under the following conditions [7]:

- Unavailability of hardware or software component which is external to the system during execution
- When there is a blocker not allowing further testing
- A specific holiday period which shuts down testing

Testing is resumed under the following conditions:

- When the external components are available
- When the blocker has been removed and a fix is implemented successfully, and testing team is notified
- When the holiday period has ended

2.3. Test Completeness

Testing will be considered complete when the following conditions are satisfied:

- When Developers agree that testing is complete, the system is stable, and agree that the system meets functional requirements
- Script execution of all test cases in all areas have passed
- Automated test cases have in all areas have passed
- All priority bugs have been resolved and closed
- Each test area has been signed off as completed by the Test Lead

3. Test Deliverables

During the project, the following items have to be tested:

Deliverable
Documents
Executive Summary
User Stories
Data Flow Diagram
Class Diagram
Sequence Diagram
Entity Relationship Diagram
Deployment Diagram
Security Plan
Testing Plan
Reports
Test results report
Test Final Report - Sign-Off

3.1. Responsibility Matrix

Based on the Responsibility Assignment (RACI) matrix[8], each team member has been assigned deliverables for a project.

- Responsible: person who performs an activity or does the work
- Accountable: person who is ultimately accountable and has Yes/No/Veto

- Consulted: person that needs to feedback and contribute to the activity
- Informed: person that needs to know of the decision or action

	Naresh	Pallavi	Soma Sunder	Shikhar	Shruthi	Shuang	Ashwin
Executive Summary	C, I						R, I
User Stories	A, I			C	R		I
Data Flow Diagram	I		R	I		C	
Class Diagram	R	C		I	A		
Sequence Diagram		R		I	C	A	
Entity Relationship Diagram		A		R	I		C
Deployment Diagram	C		A		I	R	
Security Plan		A	R	C			I
Testing Plan			C		R	A	I

R = Responsible, A = Accountable, C = Consulted, I = Informed

4. Acceptance Criteria

Test Case ID	Test Scenario	Test Case	Pre-condition	Test Steps	Expected Result	Post Condition	Actual Result	Status (Pass/Fail)
T-01	Add Tenant Information	Validate Tenant Name, Phone Number and Deposit Amount	The employee must be logged in	1. Enter Tenant Name and validate for characters only 2. Enter Phone No and Validate for numbers only 3. Enter Deposit Amount and Validate for float numbers only	Name, Phone No and Deposit Amount should be saved successfully in the database.	Name should have only characters (A-Z a-z) Phone No and Deposit Amount should have only number (0-9) Deposit Amount should have only number (0-9) up to 2 decimal points		
T-02	Save button should be enabled after validating the the tenant information	After entering tenant information, save button should be enabled if all the fields are validated as per their format	No fields(name, Phone no and Deposit Amount) should be blank and field should match desired format	1. Enter Tenant Information (Name, Phone No and Deposit Amount). 2. Once Save button is enable, click on it.	After Successful validation, save button must be enabled otherwise it must remain disabled.	Confirmation Pop-up must show if the data is saved correctly or an Error Pop up message must pop up		
T-03	Edit existing tenant information	After existing tenant found using tenant id, Tenant information - Name , phone no and deposit amount can be edited	Tenant record is found using tenant id	1. Find Tenant record using tenant id.2. Once found, the tenant name, phone number and deposit can be edited.3. Once validated, if save button is enabled, click to save the details4. If successfully, show confirmation otherwise show error message	Once Tenant record found, user should be able to edit the information and it should be validated as per their format. Once validated , save button should be enabled and after clicking , show confirmation if successfully saved otherwise show error message	The new information should be reflected on the tenant record		
T-04	record checks in batch mode	After entering tenant information, number of checks pending should be listed	rent information is already entered	check if the previous months' rent have been paid	the output should have the rent based on months in grid structure	Tenant details with updated rent information can be updated		

T-05	Record information of payments	The test should tell us the source of payment for a tenant and the amount that is due for rent	The payment field should be filled with some amount	1. Enter Tenant information and it should tell his monthly rent 2. The rent should be compared with the source of income. If the source of income is HAP, parents or non-personal income, the difference should be shown in the amount due	Success of this step shall determine the correct amount which is due for a tenant.	Correct rent amount shall be displayed		
T-06	Add propety and unit information	Validate property name ,unit name,Address and Select the property type from drop down	Employee must be logged in	1. Enter propert details such as Name,Address and Unit details 2. Click on "Save"	Propery information is successfully saved and is displayed to the user.	On successfull save, confirmation message should pop up , otherwise an error message will pop up		
T-07	Track the property and unit information	Search property or unit by property name or address	Employee must be logged in and property information must exist	1. Search property and unit information by Property name or Unit 2. Display the matched property 3. If not matched, show error message	Matched property should display all property and unit information	Property and unit inforamtion can be printed and saved on file.		
T-08	Add lease informaiton	Validate tenant name, property name, unit number, rent amount	The employee must be logged in	1. Enter Tenant Name and validate for characters only and the name is existing in the database 2. Enter Property name and Validate for characters only and the name is existing in the database 3. Enter Unit number and Validate for numbers only and the number is exsiting in the database 4. Enter Rent Amount and Validate for float numbers only	lease information like tenant name, property name, unit number, rent amount have been saved successfully in the database	Name should have only characters (A-Z a-z) Unit number should have only characters(A-Z a-z) and number(0-9). Rent Amount should have only number (0-9)		
T-09	Edit existing lease informaiton	Find existing lease information using tenant id, Tenant Name, or unit number, all of the lease information can be editted and save successfully	The employee must be logged in, Lease record is found using tenant id	1. Find Lease record using lease id or tenant id or unit number. 2. Once found, the lease information like rent amount can be edited. 3.Once validated, if save button is enabled, click to save the details 4. If successfuly, show confirmation otherwise show error message	Once Lease record found, user should be able to edit the information and it should be validated as per their format. Once validated , save button should be enabled and after clicking , show confirmation if successfully saved otherise show error message	The new information should be reflected on the lease record		
T-10	Make monthly rent adjustments	Enter valid rent amount adjustement	Employee must be logged in	1. Employee naviagtes to the rent details page for a particular tenant 2. Employee can view the monthly owed rent amount of the tenant 3. Employee adjusts the rent amount	The employee is able to adjust the rent amount and make monthly adjustments. The adjusted changes are			

				for the selected tenant 4. Employee clicks on 'Save' button to save the changes	immediately reflected and saved in the system			
T-11	Make monthly rent adjustments	Enter valid rent amount adjustement	Employee must be logged in	1. Employee naviagtes to the rent details page for a particular tenant 2. Employee can view the monthly owed rent amount of the tenant 3. Employee adjusts the rent amount for the selected tenant 4. Employee does not click on 'Save' button	The adjusted changes will not be saved to the system			
T-12	Make monthly rent adjustments	Enter invalid rent amount adjustment	Employee must be logged in	1. Employee naviagtes to the rent details page for a particular tenant 2. Employee can view the monthly owed rent amount of the tenant 3. Employee adjusts the rent amount for the selected tenant and enters an invalid amount 4. Employee does not click on 'Save' button	The system will show an error stating that 'The adjustment amount in invalid'	Employee should be able to close the error pop up message		
T-13	View rent payment sources	View valid multiple rent payment sources	Multiple rent payment sources exist	1. Employee naviagtes to the rent details page for a particular tenant 2. Employee can view the monthly owed rent amount details of the tenant 3. Employee clicks on the 'View payment sources' button	The system will display all the multiple sources of the payment such as : federal subsidy, multiple tenants each paying a portion, parents paying part of the rent, HAP subsidy			

T-14	View rent payment sources	View multiple rent payment sources	Multiple rent payment sources do not exist	1. Employee naviagtes to the rent details page for a particular tenant 2. Employee can view the monthly owed rent amount details of the tenant 3. Employee clicks on the 'View payment sources' button	The system will display just a single source of payment for the selected tenant			
T-15	Track outstanding amount	Track outstanding rent amount by month	Employee must be logged in	1. Employee naviagtes to the reports section 2. Employee selects 'Generate amount due report (last month)' 3. Employee clicks on the 'Submit' button	System will display the list of tenants and the outstanding amounts for the last month			
T-16	Track outstanding amount	Track outstanding rent amount for last 60-90 days	Employee must be logged in	1. Employee naviagtes to the reports section 2. Employee selects 'Generate amount due report ' and selects the period as last 60 or last 90 days 4. Employee clicks on the 'Submit' button	System will display the list of tenants and the outstanding amounts for the requested period			
T-17	Track outstanding amount	Track outstanding rent for unit	1.Employee must be logged in 2.Unit is leased	1. Employee naviagtes to the reports section 2. Employee selects 'Generate amount due report ' and selects the unit for which he/she wants to view the details 4. Employee clicks on the 'Submit' button	The current monthly payment history by unit will be displayed showing rent due, payments made and balance due			
T-18	Generate income reports	Generate income reports by month	1. Employee must be an administrator 2. Employee must be logged in	1. Employee naviagtes to the reports section 2. Employee selects 'generate income report' and selects the month for which he/she wants to view the report 4. Employee clicks on the 'Submit' button	The income report for the selected month is displayed. The income values generated in the report are correct.			
T-19	Generate income reports	Generate income reports by month	1. Employee is not an administrator 2. Employee must be logged in	1. Employee naviagtes to the reports section	Employee is not able to view the income reports			

T-20	Generate income reports	Generate income reports by tenant	1. Employee must be an administrator 2. Employee must be logged in	1. Employee naviagtes to the reports section 2. Employee selects 'generate income report' and selects the tenant for which he/she wants to view the report 4. Employee clicks on the 'Submit' button	The income report for the selected month is displayed. The income values generated in the report are correct.			
T-21	Track outstanding amount	Generate income reports by tenant	1. Employee is not an administrator 2. Employee must be logged in	1. Employee naviagtes to the reports section	Employee is not able to view the income reports			
T-22	Track vacant properties	Generate list of vacant units in a property than can be assigned to new tenant	1. Employee must be an administrator 2. Employee must be logged in.	1. Employee logs in to the system as administrator 2. Employee clicks on button to retrieve list of vacant units in a property	Employee is able to view the list of vacant units that can be offered to new tenant	Most updated list of vacant units shall be displayed		

Each document that has been designed must meet the following acceptance criteria:

S.No	Document Name	Testing Acceptance Criteria	Sprint
1	Executive Summary	It provides a brief summary of the system's requirements, usage and functioning. Ensure that all the requirements are captured effectively	1
2	User Stories	It provides a simplified description of the requirement. It should have acceptance criteria along with the descriptionn of functionality in a more detailed manner	1
3	Data Flow Diagram	It is a graphical representation of flow of data through an inforamation system. Ensure that the process aspects are modelled	2
4	Class Diagram	It shows the system's classes, their attributes, methods, methods and relationships among objects. Esure that	2
5	Sequence Diagram	It shows the interaction between various objects in a timely manner	3
6	Entity Relationship Diagram	It shows how entities are relateed to each other, their attributes and multiplicity	3
7	Deployment Diagram	It shows how the software components are placed on hardware components	3
8	Security Plan	It shows how the data and the entities containing it are protected	4
9	Testing Plan	It shows how the testing of the various functionalities of the system is performed	4

5. Plan Approvals

Name	Signature	Date
Soma Sundar	**S.S**	12/04/2018
Naresh Choudhary	**N.C.**	12/04/2018
Pallavi Tiwari	**P.T.**	12/04/2018
Shuang Li	**S.L.**	12/04/2018
Shikhar Chandra	**S.C.**	12/04/2018
Ashwin Bhide	**A.B.**	12/04/2018
Shruthi Shetty	**S.S.**	12/04/2018

References

[1] Parashchenko, R. (n.d.). Software testing test plan templates. Retrieved from https://strongqa.com/qa-portal/testing-docs-templates/test-plan

[2] Software testing. (n.d.). Retrieved from http://selenium-beginner.blogspot.com/2013/01/test-plan-for-manual-testing.html

[3] Amadi, T. (n.d.). PROPERTY MANAGEMENT SYSTEM. Retrieved from http://www.academia.edu/18928139/PROPERTY_MANAGEMENT_SYSTEM

[5] What is Performance Testing and Types of Performance Testing? (2016, December 18). Retrieved from https://www.softwaretestingclass.com/what-is-performance-testing/

[6] Security Testing. (2018, July 13). Retrieved from http://softwaretestingfundamentals.com/security-testing/

[7] Vinod, Ramachandran, & Mackerras, C. (2018, June 07). What is Acceptance Testing (A Complete Guide). Retrieved from https://www.softwaretestinghelp.com/what-is-acceptance-testing/

[8] Test Plan. (n.d.). Retrieved from http://www.onestoptesting.com/test-plan/resumption-requirements.asp

[9] How to Do RACI Charting and Analysis: A Practical Guide. (n.d.). Retrieved from https://www.projectsmart.co.uk/how-to-do-raci-charting-and-analysis.php

Security Plan

This security plan template is adapted from Info Security Plan Template [1]

Security Plan **System Name: Property Management System**	
Application Name	Property Management System
Brief Description	This application helps to automate the rent tracking process. It captures the amount due, the check amounts and provides historical information
Application Data Owner(s)	Property Manager
Business Area/ Department	Property Management
Summary of Hardware / Software	Application Server (On-Premise) SUSE Linux Enterprise Server *x*64 (64-bit) Database Server (On-Premise) PostgreSQL 9.5.3 Software: Nagios
Overview of Security Requirements and Processes	• The system shall be secured from any unauthorized access by using efficient Access Management • The system shall generate a log report in case of any unauthorized activity or system failure. • Each logged user will have a secure session.

	● The server shall have the latest Anti-Virus and should be updated at regular intervals. ● The system shall take maximum 10 Seconds for a screen to load, otherwise, the connection will be dropped. ● The system shall record all information with the timestamp, user and comments. All old values will be moved to a history table ● Hashing and Encryption of sensitive information in the database.
Legal / Regulatory Requirements for Security / Privacy	Property Management Laws by the state which affect the HAP subsidy.
Confidentiality Requirements / Data Classification	**Internal** - This information can be accessed and modified by property owner only. **Confidential** - This information is owned by the property owner but cannot reveal this information to anyone except the concerned individual **Public** - This information is allowed for public viewing. ● Tenant Information – Internal ● Property Information – Public

	● Rent and Expense Information - Internal
Administration, Roles & Responsibilities for Security Functions	Property Owner ● Responsible for adding/modifying tenant, property and rent information ● Records the checks into the system. ● Approves system access. ● Generates monthly rent report
Security Logging and Monitoring	**Nagios**, a server monitoring software, will be installed on both Application and Database Server for server monitoring and logging When the Server reaches 80% utilization or there is a server outage, a notification will be sent to the property owner. A simple interface is used to access any server log and generate reports.
Security Training	Employee IT Security Training ● This training will help administrators understand the server logs, understand the emerging information threats and data privacy. Initial Application Training ● This training will help new employees understand the working of the application

Security Testing	At the Server Level, below are the testing requirements ● Penetration Testing ● Load Testing ● DoS Testing ● Vulnerability Scanning At the Application and Database Level, below are the testing requirement ● URL Manipulation ● Unauthorized Data Access ● Cross-Site Scripting ● SQL Injection
Infrastructure, Telecommunications, and Environment Security Components	● Operating System on the server will be updated with the latest patches and Anti-Virus ● Each session on the application is secured via a secure session. ● All codes and quality documents are version controlled and kept at a single location. ● Nagios monitoring software will help detect any unwanted intrusion. ● Authenticate every session in which QuickBook updates the system to ensure legit data is entered into the system without getting leaked to a third party

Backup and Disaster Recovery Requirements	Backup Requirements ● Daily Database backup ● Weekly Application Code backup All backups are stored on tapes. This is retained for a week. These tapes are used to recover data in case of a disaster or system failure.
Remote Access Requirements	The application can be accessed remotely only through VPN (Virtual Private Network). These credentials are provided only to the property owners.
Physical Security	● Identify the vulnerabilities in the area that affect the server. ● The server can be accessed only by the property owner and personnel authorized by the property owner ● Data backups are to be checked daily by the system administrator or approved personnel by the property owner.

Reference:

[1] - "Annex C Info Systems Security Plan Template." *Housebuilding: Permanent Dwellings Started and Completed, by Tenure*, 7 May 2012, data.gov.uk/data/contracts-finder-archive/contract/576979/

SCRUM Essay

We chose the scrum process so that we could provide not only a rich user experience but also a highly efficient and robust system to the user. The process helped us deliver quality deliverables and bring out the best team performance.

The creation of the product backlog helped in prioritizing the deliverables and provided a neat way to release each deliverable. This helped in delivering the most significant deliverable first. User stories were used to describe each deliverable as it made it easier to understand the system's features from the customer's point of view. To capture and track all the user stories, we used APA[1] Scrum board. As we improvised our initial deliverables, it was easier to work on the next items on the product backlog. This gave the team members an opportunity to clarify the stories and estimate the story points.

Our sprint planning took up a big chunk of our time, but this eventually helped us estimate our sprints more accurately. This aided in breaking down complex features into smaller, manageable tasks. During the planning, the user stories in the product backlog were discussed and debated in depth. This gave a chance for each member to raise their opinion or questions, so that minimum roadblocks are encountered during the sprint. The planning also helped in assessing the technical competency and assigning deliverables accordingly. By the end of each planning, team members were better prepared to work towards their deliverables.

Our daily standups proved to be beneficial for several reasons. Each member of the team was aware of the progress of the sprint. The effective communication between team members provides transparency to remove any kind of blockers in the initial phase and focus on the outcomes rather than just getting the work done. During standups, each member was required to give an update of

their deliverable. Hence, a sense of responsibility was instilled in each member to bring out quality results. After the standups, the scrum board was updated and sent to all team members to ensure everyone is on the same page.

Sometimes the standups would lead to certain conflicts among the members while discussing their blockers. For example, while designing the class diagrams, many discussions erupted during the standups, delaying the entire process. The scrum master helped in bringing back the ultimate focus of the team and resolved their conflicts or removed any blockers offline. This ensured respect for each team member's time.

Sprint demo helped the team members showcase their work, making them feel that they have created some value. Incremental feedback provided the team with better ideas and improve their deliverable after each sprint. The feedback received during the demo helped us understand the bigger picture of the property management system that led to changes in class and ERD diagram.

Sprint Retrospective gave a chance for members to look back and identify the positive and negative aspects of the sprint. An open discussion was held, and team members were encouraged to speak about their experiences. The product owner highlighted the feedback provided by the customer after each sprint demo which helped the team to improve their deliverables at the early stages. This has also helped in reducing many project risks such as ill-defined scope, change management and stakeholder's engagement.

With the scrum process, we came up with well-defined roles and responsibilities that led to smoother iterations and fewer conflicts. It has helped the team to communicate effectively to drive development and produce a world-class property management system.

References:

[1]-"Simplified Online Scrum and Kanban Tool." *Apa - Simplified Online Scrum and Kanban Tool*, apascrum.com/.

Agile Essay

We will introduce four kinds of Agile methodologies which are Scrum, Kanban, Extreme Programming and Lean Framework. After contrasting and comparing these four methodologies, we will introduce how they would have worked differently for our project.

Scrum

Scrum is a framework within Agile methodologies which involves continuous communication between the stakeholders. A Scrum team consists of a Scrum Master, Product Owner, and a group of developers and testers. In some cases, even the design team is part of the Scrum. The entire development life cycle is divided into sprints. Each sprint has certain tasks that contribute to the development of the final product. Scrum has rituals like daily stand-up during which the current status of each task is monitored. There are also artifacts like the burnt down chart and velocity chart which helps in keeping track of the number of tasks that are completed, or the number of sprints that is left before the completion. The product owner is the bridge between the customer and the development team. He gathers the requirement and discusses it with the developers to figure out the time that would be taken to complete each user stories.

Scrum has been immensely helpful in our project. All the team members have been continuously involved during the life cycle of the project. They have shared timely feedbacks which have helped in the continuous improvement of the project. Even if there are any gaps in the understanding, the product owner ensures that he or she is on the same page with the customer to clarify the requirements. One of the most essential aspects has been, that one team member does not have to depend on others for a task to be completed. The tasks or the user stories are picked up by the engineers and completed based

on the effort number associated with those tasks. This ensures that the workflow is maintained consistently.

Kanban

Kanban is a framework within the Agile methodology. It was developed by the Toyota corporation to undertake swift manufacturing activities. Kanban is almost the same as Scrum but differs in how the development process flows. There is no daily standups or continuous communication between the team members. At the same time, there is a Kanban board which tells us about the different phases in which the development process lies. The different phases can be To-do, In Progress, Test and Done. Every phase has a limit in terms of the number of tasks it can hold. One of the advantages is that it prevents building up of overhead. Once a limit has been reached for a phase, the engineers working on it would be notified that the overhead has been reached and the tasks should be moved to next phase soon. Until the limit is cleared, any additional task cannot be pulled in from the backlog. Another similarity with scrum is, that the tasks are pulled from backlog based on priority. The priority is decided at the beginning of the project. Unlike scrum, there is no daily standup during each sprint to decide the user stories that need to be taken up.

In our project, Kanban might not have been suitable because Kanban is mostly suited for small teams and the priority is decided at the beginning of the project. Unlike scrum, a set of user stories are not decided at the start of each sprint. Instead, the tasks are pulled from the list based on priority. This has a chance of the feature hitting a bug. Although it's a fast-paced approach, it creates overhead for testers and can sometimes lead to a delay in the delivery of the project.

Extreme Programming

Extreme Programming is another framework within the agile methodology. Extreme programming is mostly beneficial for small or mid-sized teams. This framework keeps the customer and developer in the center of the project delivery. Although there is continuous customer interaction, there is no scrum master involved. The developers themselves swap the role of interacting with the customers. Extreme programming has multiple facets like – pair programming, code review, unit testing and respect for team members among others. In pair programming, two developers work together on a single computer most of the times. One acts as the navigator and the other acts as the observer. One of the advantages is that the code review also keeps happening side by side and hence the time taken for separate testing is saved. The navigator and the observer can switch roles based on their proficiency of the technology stack they are working on. Extreme programming also focuses on unit testing. In this, instead of testing the complete code, a small critical unit is tested against multiple scenarios. Multiple small units are tested keeping in mind the criticality and the priority of their roles in the entire product. Extreme programming also focuses a lot on respect for each of the team members. Since the developers also act as the point of contact for the customers, their views and opinions are not questioned.

Extreme programming is a good framework to undertake but only in those scenarios where the customer requirements change frequently. The customer requirements might change depending on the changing market conditions. Hence, it might not have been helpful in the property management system. This is because once a set of requirements have been collected, there would hardly be any change, apart from the acceptance criteria that the customer is looking for. There won't be many changes based on the changing market scenario.

Lean Framework

Lean is another framework which is used within Agile methodologies. It was originally used in manufacturing business, but later became part of Agile in 2001. The main goal of lean is to optimize the process and minimize the wastage of the resources. More than a framework, it's a process to optimize the development methodology. It's a people-driven framework, where the developers and engineers are at the forefront of the entire process. One of the major differences between Lean and other agile frameworks is the number of features that are developed at one point of time. In frameworks like XP, Scrum, and Kanban, developers might be working on more than one feature at a time, whereas in Lean, the developers might not be working on a new feature until the current feature is complete. In most of the scenarios, it might happen that one developer might be working on only back-end before moving to the front end. Lean focuses a lot on the engineers and the way the product is delivered. This is called as a 'System' in Lean.

Lean framework, if used in our project, could have delayed the deliverable. This is because two or more features could not be developed at the same time. Also, the project has multiple members, and if not all the members are not allocated multiple features, this could lead to the non-optimal utility of resources.

In summary, the four methodologies all have different characteristics and whether are they suitable is based on the size and type of the project. For our project, we think the Scrum methodology is the most suitable one.

Waterfall Essay

We need to go through six phases to get our project finished, which are gathering requirements and documentation, system design, implementation, test, product delivery and maintenance.

Gathering Requirements and documentation

Firstly, we need to gather the requirements and document all of them comprehensively. For this project, we need to talk to Ms. Mywick to get her specific and well-defined requirements. The requirements must be clear and complete, and each member of the team must know them well. The output of this phase is the requirement analysis document, which should be distributed to the team.

System Design

Based on the established requirements, the team begins the designing process. No code will be generated during this phase. The team will decide what kind of architectures will be used. Also, the team will use UML diagrams like class diagrams, sequence diagrams and deployment diagrams to figure out the critical parts of the system. The team will also decide on what programming language will be used and the hardware requirements.

Implementation

All the coding work will be finished using all the documents created during prior phases. Programmers will review the documents, get information they needed from previous phases, and create a functional product. In our project, the programmers can implement the code in small pieces to realize different functionalities. All of the code will be integrated at the end of this phase.

Test

When all coding is done, the testing will begin based on the test plan. Testers will test the property management system methodically and report bugs and problems. If some serious and fatal problems arise, the team must figure it out and go back phase to make modifications.

Deliver the product

After all the testing work, the product is ready to be delivered and released.

Maintenance

When the product is released and used by the customer—Ms. Mywick, she may find something that is not as expected. We must make modifications based on her feedback. Again, any big issues require return to phase one.

I don't think the waterfall methodology works well in this project since it's not efficient. There are three reasons. Firstly, only when the prior phase has been completed can we begin the next phase. If we are stuck in one phase, it will result in the severe project delay. Secondly, we also cannot go back to the previous phases unless we start from the phase one again. But in this project, some of the requirements are not clear and easy to change. If we use the waterfall methodology and deliver the product to get feedback in the end, we must start from the phase one again to make changes and it really wastes time. Even if we find some big problems during the process, we also can only go back to the phase one, so it's really inflexible. Last, we must do the complete and proper planning before we start to do coding, but actually it's hard to predict everything forehead.

In summary, we can use the waterfall methodology for our project but it's not suitable and inefficient.

Scrum Master Essay

Sprint 1

I was the scrum master for the sprint 1. Being the first sprint, it was challenging to hold scrum meeting and achieve daily goals. Team was new and took a while to bond with each other. But after two or three daily meetings, everyone formed a professional bond and all were familiar with the case study. Starting in a slow pace, team worked harder and proactively completed task. Holding meetings with a new team with vague roles, my experience as a scrum master was challenging but intense learning experience providing an opportunity to understand why scrum is a successful software development approach.

We started with Sprint planning where our goal was to create product backlog. My focus was to complete the product backlog as soon as possible so that individual task could be assign to scrum team members. This empowered team to work in parallel and complete deliverables independently. Why daily stand ups are only 15 minutes? During our meetings, I was able to figure out why stand ups are only 15-20 minutes. Starting with 'in which class payment list must be stored', after an hour of brainstorming we concluded the discussion with a decision to make a whole new class diagram. But in the next stand up a team member came up with a simple fix which resolved all our issues. Conclusion? As a developer we tend to believe our ideas to be perfect and we try to stick to it. But it is the responsibility of scrum master to understand the same and stop a conversation going in irrelevant direction. For example, I tried to keep the conversation short if it was not helpful to others. Conversely, topics which assisted team in future task were kept longer on table.

Lastly retrospective helped us to conclude team performance and helped me discern my shortcoming and strengths as scrum master. Overall it was a great experience managing people and creating a learning environment.

Sprint 2

The role of Scrum Master for Sprint-2 was carried out by me. After the Sprint-1 retrospective, the team was aware of the shortcomings and the things that needed to be improved going forward. The main challenge for me was to perform planning of the upcoming sprint. After the end of the first sprint, an estimate of the number of story points that the team could complete was established. The strengths and weaknesses of each team member were identified, and the assignment of stories was done accordingly. My major goal was to deliver a complete slice of the product by covering many functionalities to get feedback from the Professor early into the project lifecycle.

Most of the user stories along with the acceptance criteria were finalized by the end of the first sprint. The user stories covered as a part of the Sprint Backlog included creation of the Data Flow Diagram, Class Diagram, Sequence Diagrams and Entity-Relationship Diagram. The major hurdles faced as a Scrum Master was organizing daily stand-up meetings and making sure that the team adheres to deadlines.

Daily stand-ups were supposed to be completed in less than 15 minutes, but the amount of time required for coordination between the team members took a lot of effort. Each member had a different work schedule and deciding on a fixed meeting time was difficult. There were instances when a team member arrived late, and the entire team had to wait for the meeting to start. To tackle such situations, we decided to conduct video conference calls when it was not possible for a teammate to attend the standup in-person.

The next challenge was making sure that the project deadlines are met. Some team members needed time to come up to pace with the project deliverables as they had no experience of developing UML diagrams. I decided to pair an experienced team member with a novice to counter this issue. This strategy worked as the senior member was able to provide guidance and help the

inexperienced member in developing artifacts. In addition, new ideas were introduced, and a different perspective was brought in.

All in all, the experience of working in an Agile team and acting as a servant leader for a self-organizing team was enriching. The main lesson learned while acting as the Scrum Master was to be a facilitator for helping the team achieve its goals by removing hurdles and ensuring meaningful communication across all the stakeholders.

Sprint 3

I was the scrum master for sprint 3. The feedback received during sprint 2 demo was extremely beneficial to the team as it provided a broader understanding of the property management system.

During sprint planning, a lot of debate took place regarding the class diagrams as the feedback provided during the demo needed to capture some major features that we had missed previously. It was a challenge to finally decided upon the deliverables to be taken up for the next sprint as the discussion on the class diagram acted as a blocker to complete other deliverables. Finally, I acted as the mediator and decided the sprint-3 backlog. The responsibilities were clearly stated, with each member confirming to them respectively.

Our last sprint could not cover two user stories which were carried over to sprint 3. This added some pressure on the deadlines and hence we were lacking behind in some of our deliverables. My major challenge was to make sure that we adhere to the timelines and are progressing on the deliverables, so that we can demonstrate most of the deliverables to receive feedback and make sure we are not missing out on any important details. I got members to work in teams of two. This way one could design and the other could review each deliverable. Also, the standups helped me

understand the status of each deliverable, along with any kind of blocker faced by each team member. With this, I was able to add more people to work on the difficult tasks, once they had completed their respective deliverables. This way the deliverables were developed more efficiently, allowing team members to share responsibilities.

I involved the product owner to help guide and motivate the team to work harder on their deliverables and deliver quality results. During the standups, many times team members would break into a discussion about the class diagrams that would hold up the standup, hence making the team members wait unnecessarily. I had to ensure I get back the focus of the team and take the discussion offline.

All our deliverables were collected and saved on the drive. The final presentation was prepared and delivered by the team. The feedback received during the demo helped a great deal to understand some major aspects of the class, ERD and sequence diagrams. During the sprint retrospective, we discussed some technical glitches faced during the demo and how we should prevent them in future. Some of the deliverables such as the deployment diagram was not clearly demonstrated. Hence, we need to slow down our pace during presentations.

Our progress on the deliverables was on track as the whole scrum process was extremely helpful in bringing the team closer and building the team spirit.

Sprint 4

For the group project of the Mywick property management case, I was the scrum master for the last and final sprint number 4. By the end of the third sprint we had completed almost all the deliverables especially the class diagram, user stories, sequence diagram, entity relationship and

deployment diagrams. After our third demo, we received a lot of suggestions for improvement areas in almost all our deliverables. During our demo, we also faced few unexpected glitches. As a result, the team morale was low.

My job as a scrum master for the last sprint was to conduct one sprint planning and two stand up meetings. The main job for me in this sprint was to get all the deliverables completed on time, reviewed by everyone on the team and have four scrum master essays ready by the end. Since we needed to submit the project after this sprint, it was necessary to get everyone back on track and increase their pace.

In the sprint planning meeting, I had to ensure that after all the issues we faced, the team needed to get together and be on track in terms of deliverables. In the retrospective of the previous sprint, we had touched upon how we were lagging and needed to get things done efficiently. Thus, as scrum master, it was my duty to motivate the team to get the tasks done for the current sprint. We needed to work on a lot of changes and required all the members to participate. During the sprint planning, I had to ensure that all the deliverables that needed modifications were assigned to some member. Additionally, the testing plan was not completed in the last sprint, which was also assigned to be completed.

The major challenge that I faced being a scrum master was to co-ordinate meetings between the members. There were occasions when the product owner and few members needed to meet to discuss blockers. This was difficult as everyone had busy schedules. Few members in the team have part time jobs making it challenging to set up meetings among them.

Another challenge faced during the stand-up meetings was that not everyone was physically present during the stand up. For this, the person(s) would join remotely on video call. At times

there were network issues, and this caused delays and extended the time of the meetings. Due to these network problems, the videos had to be re-recorded several times, which was frustrating.

All in all, it was a great learning experience being a scrum master. I learned how to manage and motivate the team to work together to produce valuable product deliverables. I learned how to manage and help people to effectively communicate with each other. It was a good opportunity for me to understand the role of scrum master in agile teams.