PROJECT TITLE	**Residents Management System**		
COMPANY NAME	**Southern Management Corporation**	LOCATION	**Prince George County, MD**
PROJECT MANAGER	**Sakshi Sharma**	DATE SUBMITTED	**7th March, 2020**
AUTHOR	**Pratik Pandey, Manasvi Prasad, Manas Mishra, Yash Srivastava**	VERSION	**1.0.0**
PROJECT BEGIN DATE	**10th February, 2020**	END DATE	**16th April, 2020**

Table of Contents:

EXECUTIVE SUMMARY

Overview

The undertaken project is for Southern Management, one of the largest property management firms in the Mid-Atlantic region, founded in 1965, and is aimed to benefit the firm on multiple levels. It is especially aligned with the core of the firm's mission : Customer Care and Commitment to Service. As the firm has pledged to treat all residents with an uncompromising level of customer service, regardless of the size or location of their apartment home. The functions impacted or added through this project are aligned perfectly with the Corporation's pledge to provide the residents with convenient methods to express concerns, pay rent, and use services and amenities.

Need of the Project

The project has been designed after careful collection of customer feedback and aims to resolve issues that impact the customers primarily. Through several requirement elicitation techniques, the most primary functions that not only impact the firm's customers, but also the firm's employees. The project targets identified spots for improvement that ease the process for both existing and potential residents in terms of maintenance requests, finding flatmates, transfer requests, notification systems and so on. The legacy system lacks functions that impact customer experience adversely and result in decreasing trust and adverse behavior. Duplicate rental applications also result in unnecessary increased load on the system and confusion with demand analysis and waitlist accuracy. The need for the Project arises out of targeting to resolve these improvable interactions of the users with the system.

Solution to the problems

The solution lies in improvement of existing functions of the legacy system, while adding a few new ones like addition of an interaction portal between existing and potential residents for finding flatmates and easy buy and sell. This increased coordination would result in a direct revenue boost as the number of vacant flats decreases. Online in-house transfer requests and notification system increase customer experience as convenience is key with today's fast paced lifestyle. The project is a win-win for all parties involved. The triple constraints have been analyzed in detail and the project is feasible. Also, analysis and comparison of cost-benefit between Cloud-based and Hybrid models tells us that Cloud-based is better suitable.

Value to the firm

The project intends to bring to the firm several tangible and intangible benefits perfectly in sync with the mission of the firm. Like any other well-designed system, these functions will minimize the cognitive load on the users. The improved customer experience and network of customers through the new portal is bound to bring a better market share and brand recognition. The improved network of residents is expected to attract more property owners due to cross side network effects. Better notification system is aimed to reduce complaints and customer care calls. The removal of duplicate applications will reduce IT costs and load of the database. The project will reduce expenditure and could reduce management man hours required saving costs for more Fair Pricing and invest in product betterment.

Conclusion

This project aims to strengthen and develop ties with new partners and is a result of a thorough Feasibility study, Planning, Analysis, Design and Implementation phase with elaborate DFDs ERDs and Design architecture for proper modelling of the Database. Fact Findings have assisted in development of more useful processes. The finalized functions are a new Application/In-House Transfer Request, a new Notification System, a new Roommate search portal, Removal of duplicate applications and development for a transparent waitlist. The project is of enhancement in nature and the Planning, Analysis, Design and Implementation phases take 19,20,10 and 25 days respectively, starting on 02/10/2020 and ending on 04/16/2020.

SYSTEM PLANNING PHASE REPORT

STATEMENT OF WORK

1. Client and Industry Background

The project is an enhancement that we have proposed for Southern Management Corporation. Southern Management Corporation is the largest privately owned residential property-management company in the Mid-Atlantic region and owns and manages 77 apartment communities with approximately 25,000 apartment homes, several commercial office buildings, hotels and conference centers, ski resorts throughout the Baltimore/Washington area and in Pennsylvania. The goal of our project is to upgrade the existing residential services system in the College Park region. Being residents of a property of southern management, through our first-hand experience, we were able to identify several faulty functions or complete lack of them in the current system. Therefore, our aim is to upgrade this system to address these issues and come up with an efficient information system for better transparency and customer service.

1.1 Purpose of the Project

Since we truly understand the pain points of the users of the system and know for a fact that there is incredible scope for improvement as the Southern management resident services system could be improved with simple process changes. The project will benefit all parties involved and on multiple levels and it is designed to cover all bases in terms of both tangible and intangible benefits.

2. Scope of Work

There is significant scope for improvement and of the several functions, we identified five to be implemented, which we felt should have the most priority.

Provision for online in-house transfer request: The Resident management system still lacks an online in-house transfer request portal. Users who are looking to move apartments due to any issues with existing ones, start to look at other properties, because Southern management still doesn't allow online requests, while other properties allow them to go through everything online.

Search and find potential roommates: A portal for interaction between potential and existing residents about search for potential roommates and open spots is conspicuously absent. Existing members are forced to move out when they are unable to find roommates due to lack of an interaction portal, and they have to pay the full amount of rent. Designed to improve user experience and provide customers with ease of interaction. This will also save Southern Management from losing revenue from vacant listings.

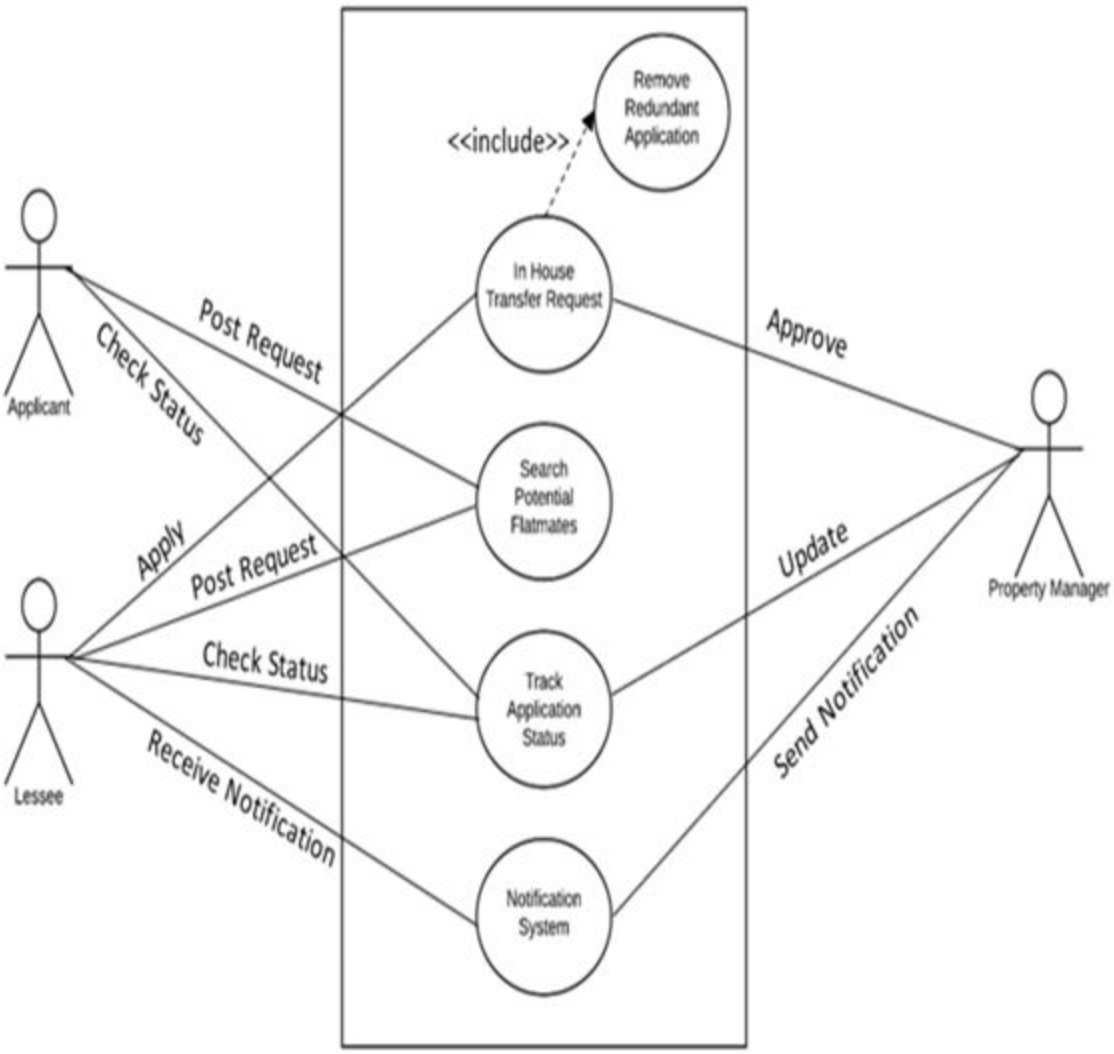
Track application status through a transparent waitlist: The waitlist is not transparent and causes mistrust between the applicants and the management. This will not affect the privacy of the applicants as we plan to show only the name of the applicants, the date they submitted the application and their standing on the waitlist, with an option to add contact details, if needed. This will help mitigate privacy issues that arise from transparency.

Improve customer notification system to provide rent alerts, service request updates etc.: We have also spotted the absence of a thorough notification system about due rent or status of service requests and any changes in the existing system.

Remove duplicate rental applications of the same apartment type: Residents and non-residents are allowed to have any number applications for the same type of apartment listing in the database. There is no restriction on this and even after finding apartments, years and years of applications are retained in the database.

Function	**Business Need**	**User Experience**
Provision for an online in-house transfer request	Reduces paperwork and data entry for the property manager	Residents can file in-house requests online without going to the resident office
Search and find potential roommates	Helps in gaining more residents and hence more houses on rent	Applicants can group up together and take apartments from the management
Track application status through a transparent waitlist	Reduces customer care calls and complaints	Applicants are made aware of their application status and waitlist
Improve customer notification system to provide rent alerts, service request updates etc.	Intimate residents for timely payments and keep them informed about requests.	Customers get timely notifications informing them regarding rent due, service requests etc.
Remove duplicate rental applications of same apartment type	Helps management in application management and house allocation.	Applicants have one common application for apartment request

Use Case Diagram



3. Project Feasibility Study:

This system proposes two technical solutions, namely, cloud-based and hybrid.

Cloud: The Southern Management System is already cloud-based. Hence, integrating the change management system is going to be much easier. To add to that, this system will provide various tangible benefits which have been discussed below in detail.

Hybrid: The alternate approach is developing a hybrid solution which would make the system more robust. Sensitive data could be stored on-premise and less critical information can be maintained on the cloud.

CLOUD-BASED	**HYBRID**
Web based solution	Web based solution.
Web based application hosted using in-house cloud service provided by Yardi	Web based application hosted in Southern Management data center
Data stored using in-house cloud service provided by Yardi Systems	Data stored using in-house cloud service provided by Yardi Systems

Comparisons between the two technical solutions have been made in the above table.

- Both the solutions are web-based. The differentiating factor between the two approaches is the server. In the cloud-based system, the server is cloud-based too. On the contrary, a hybrid model has the server situated on-premise as well as on the cloud. In this system, sensitive information is generally stored on-premise whereas the application is cloud-based. This usually propels the cost of hybrid systems.

- Another differentiating factor is the way data is stored using these two models. Currently, Southern Management uses a cloud-based system provided by Yardi systems. Yardi systems also provide in-house solutions for data storage and will be used in hybrid models.

- After comparing the two solutions, we have decided to go ahead with a cloud-based solution. This is because Yardi systems provide a bundled service for the same. To avoid migration costs, a cloud-based solution is ideal.

3.1 Technical feasibility

	Cloud	**Hybrid**
Familiarity with technology	Excellent	New
Project Size	Medium	Medium
Compatibility	High	Medium

Both the technical solutions have their own impacts on how the project pans out.

- The existing application is already cloud-based. Hence, implementing changes that are cloud-based will not require any stakeholder to familiarize themselves with the updated application. This might not be true in case of hybrid systems as the backend system might function differently and this would require familiarization from the perspective of the maintenance team.

- Since we are implementing a limited number of changes which are aimed at improving the efficiency of the system, the project is going to be relatively small in both cases. Implementing change management does not always mean that the project size is going to be small, but in this case, the project size is medium.

- Compatibility is going to be high for a cloud-based model as there would be no migrations that will take place. The new system will be based on the same architecture. However, in developing a hybrid system, compatibility might decrease when compared to a purely cloud-based system.

3.2 Cost and benefits analysis

Costs:

There are several costs involved in the execution of this change management system which can be classified into development and operational costs.

	Cloud	**Hybrid**
Development Costs		
Software development	$45000	$60000
Hardware and software	$1500	$2000
Operational Costs		
Cloud storage fees	$1000	$750
Staff training	$1500	$1500
Transactional costs	$1500	$1500

Development Costs:

- For both models, the majority of the costs are concentrated on software development. But the costs for hybrid development are much higher due to obvious reasons. Setting up the infrastructure on-premise is going to drive up the costs.
- Like software development, the cost for hardware and software would be higher for hybrid systems as cloud-based systems are much cheaper but investing in on-premises servers increase the costs to a certain extent.

Operational Costs:

- Cloud storage fees would be higher in the case of a cloud-based system.
- The training costs involved are pretty much the same in both models.
- Transactional costs for a cloud-based system would be more or less the same as hybrid systems. Both the methodologies would include planning and outsourcing but, in some cases, it could be slightly more in hybrid systems.

Benefits:

We have identified a few benefits for both the technical solutions and categorized them as tangible and intangible.

	Cloud	**Hybrid**
Tangible benefits		
Increased Sales and Revenue	Yes	Yes
Reduction in IT costs	Yes	Yes
Intangible benefits		
Increased brand recognition	Yes	Yes
Increased market share	Yes	Yes
Improved customer service	Yes	Yes

- A significant increase in the revenue from increased and consistent sales is observed for both the models and otherwise, through portals for flat mates, online in-house requests, reduced paperwork and reduced load on database.

- The mitigation of redundancy in applications will result in a lighter database.

- Better and transparent system will entice other property owners and the word of mouth will also bring more users and develop their network

- Increased transparency will consequently bring increased customer trust and a strong Customer base.

- Improving user experience, better and ease of usage functions will bring a stronger customer base for both property owners and the management.

3.3 Organizational feasibility

	Cloud	**Hybrid**
Strategic Alignment	Excellent	Excellent
Influence on Stakeholder	High	High

Based on the Cloud and Hybrid models, the organizational feasibility of the system can be measured in the following way:

1. **Strategic Alignment:**

   The new system enhances the existing system and adds new features to it. These new features are strategically aligned with Southern Management's goal to better manage apartment applications and applicants. The system will reduce paperwork and redundancy by digitalization of the process.

2. **Influence on Stakeholders:**

   The stakeholders of this enhancement are the residents, the intermediary platform, Rent Café, and Southern Management Corporation itself. The entire project management team operates under Southern Management Corporation.

   The system, when implemented, will reduce the burden on Southern Management employees and simplify various processes. This in turn will save a lot of labor hours and further the company's profits. From a stakeholder's point of view, the system will provide a high ROI (Return on Investment) as the development of the system is minimal and the benefits are high for both cloud and hybrid models. Thus, the stakeholder affiliation will increase in either of the cases.

4. Project Plan:

Location of the Project

Since the development of the Southern Management portal will be handled by their vendor, Rent Café, most of the work will be done at its headquarters in Santa Barbara, CA.

Elicitation of user requirements followed by system installation and user training will happen at Southern Management's headquarters at Vienna VA.

Project Plan First Cut

The project is going to be developed using the **waterfall development methodology**. The plan entails 4 stages which takes a total of **49 days** to complete starting on 2/10/20 and goes on till 4/16/20, which are as follows:

4.1 Planning stage:

Starting 2/10/20, taking a total of **19 days** to complete.

Tasks:

- Identification of the project requirements
- Developing a systems request
- Analysis of feasibility (technical, economic and organizational)

Milestone: Discuss and propose a plan, place a system request and run technical, economic and organizational analysis and get ballpark figures by 3/5/20.

4.2 Analysis stage:

Starting 3/6/20, taking a total of **20 days** to complete.

Tasks:

- Gather Customer requirements
- Gather stakeholder requirements
- Develop, review and approve use cases

Milestone: Study customer and stakeholder data and requirements, develop use cases which are compliant with the requirements, review and approve the use cases by 4/2/20

4.3 Design stage:

Starting 4/3/20, taking a total of **10 days** to complete.

Tasks:

- Select design strategy
- Study existing systems
- Develop use scenarios and program specifications

Milestone: The design strategy will be formulated; past and existing systems will be studied and use case scenarios and program specifications will be gathered which is needed for the new system with added functionalities by 4/16/20

4.4 Implementation stage:

Starting 4/17/20, taking a total of **25 days** to complete.

Tasks:

- Program system
- Test software
- Maintain system

Milestone: The development and testing of the system will be completed in 15 days. Maintenance and support for the new system will be provided for the next 10 days until 5/21/20.

5. Conclusion:

This statement of work recognizes the opportunities and challenges of this enhancement to Southern Management, their property owners, the residents and their brand. We appreciate the opportunity to present this and hope to add value to Southern Management through this project. We believe that the project is consistent with the goals of the firm. Once the project is acceptable to the board and gets signed for approval, the project plan that has been put into place will be implemented.

SYSTEM ANALYSIS PHASE REPORT

1. Fact Findings and Requirement Gathering

The Southern Management Corporation has long been touted as the largest residential management company in the United States. However, it has had its own issues regarding the experiences of residents and non-residents using the website to perform various functions. As part of representing Southern Management, a list of functions have been identified that can improve the efficiency of the website. Out of these, five have been shortlisted based on priority. Below is a detailed description of fact findings and requirement gathering techniques for all the functions that have been identified for execution.

Function 1:

Provision for online in-house transfer request

Fact findings: In the age of online transactions, which Southern Management has incorporated pretty effectively, in-house transfer requests were still being processed offline. This was a problem for residents who didn't stay close to the residential services offices of Southern Management.

Requirement Gathering Techniques:

Interviews: Short interviews were conducted with students staying in Southern Management communities. A top-down interview structure was followed which was open-ended as the focus was on opinions, not specifics. Interviews were documented and later sent to the interviewees for clarifications, corrections and confirmations.

Function 2:

Search and find potential roommate

Fact findings: Portal to find new roommates is absent. This often leads to people surrendering their apartments unwillingly and bearing the full rent. An interactive portal to improve residents' experience and provide them with a platform to expedite the process of finding roommates has led to the creation of this requirement.

Requirement Gathering Techniques:

Interviews: Since this requirement is aimed at improving the problems of the residents, we believe that interviews were the best way to go about it. Residents were interviewed in a similar fashion to the above-mentioned function. Users (residents) were able to provide us with clear explanations.

Questionnaires: Questionnaires were floated to the residential services teams at various Southern Management offices to understand what is needed from the service end. The added functionality will require more maintenance and gathering facts and opinions from all stakeholders is going to be helpful.

Function 3:

Track application status through a transparent waitlist

Fact findings: There is a waitlist system in place but the residents don't have access to view it. This causes mistrust and there have been a few occasions on which the waitlist hasn't been used efficiently to allot homes to applicants.

Requirement Gathering Techniques:

Document Analysis: The existing system was studied in depth. The system and its design documentation created by analysts and consultants were used to collect facts. Information was classified into current and outdated to determine its accuracy. This method helped us understand the existing system well.

Function 4:

Improve customer notification system to provide rent alerts, service request updates etc.

Fact findings: The existing application is devoid of an alert system. Although it is the responsibility of the resident to set reminders to pay bills in time, a notification system would help residents and help Southern Management manage service requests better.

Requirement Gathering Techniques:

Interviews: Residents were interviewed to collect feedback regarding the existing service request system. Their opinion on how the system should function was considered. All responses were documented and sent for confirmation.

Observations: Participated in activities performed by residents to get a hang of the existing system. This helped gather data accurately and understand what the system was doing. This technique was relatively quick to implement to gather requirements in conjunction with the interviews of residents.

Function 5:

Remove duplicate rental applications of the same apartment type

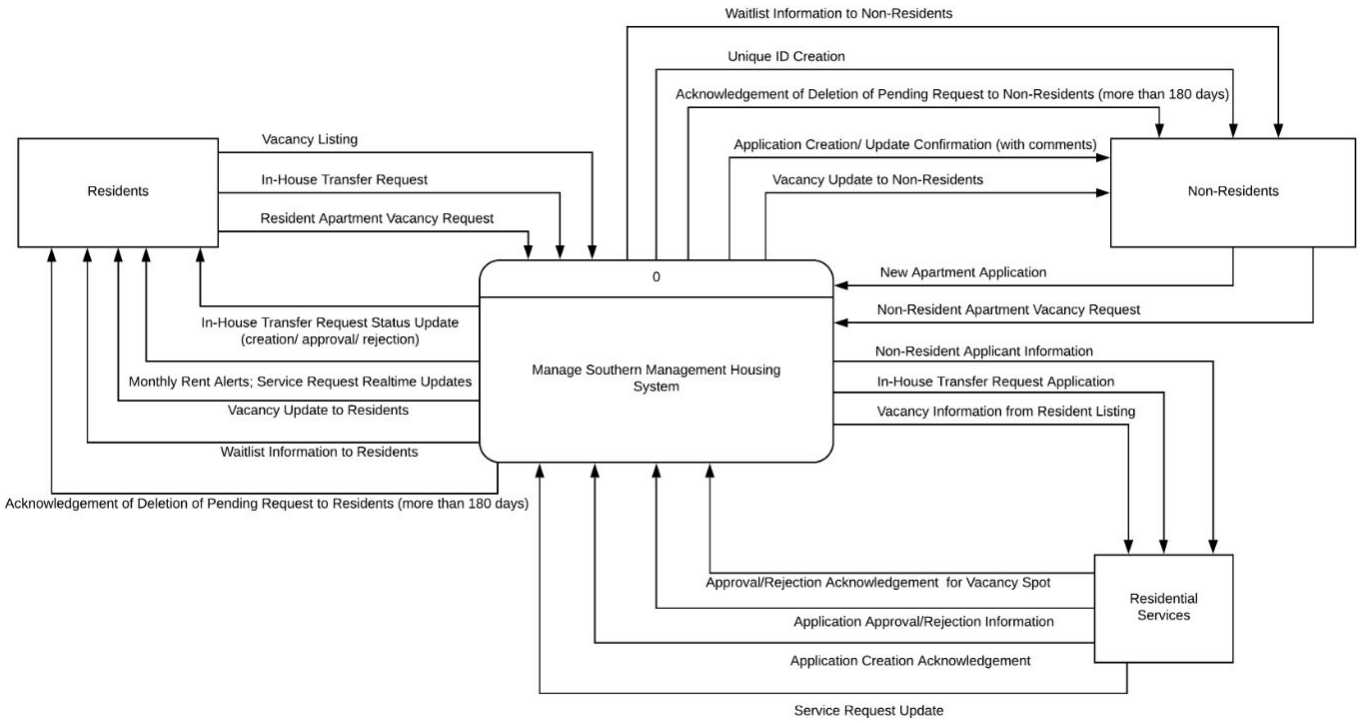
Fact findings: Unlimited applications for similar apartments were causing data storage and maintenance issues. There was no time period as to when the applications would be removed from the system.

Requirement Gathering Techniques:

Questionnaires: Out of all five functions, this function requires the most enhancement and requires very low user involvement. Hence, the approach of questionnaires was chosen. Opinions of stakeholders like system analysts were considered.

2. Proposed Data Flow Diagrams

2.1 Context Diagram



Explanation:

External entities:

- **Residents:** This external entity represents a resident that can apply for in-house transfer requests, look for space in other apartments in the community and post ads related to vacancy in their apartment on the Southern Management portal.

- **Non-Residents:** This external entity represents applicants who are not part of the Southern Management community. They fill in new applications and apply for vacancies.

- **Residential Services:** This external entity represents the residential services or in simpler terms, offices that control day-to-day activities of Southern Management. From screening applications to serving plumbing requests, everything goes through these offices and they are located in every Southern Management housing community.

Manage Southern Management Housing System is the central processing system for all data used and stored in the application creation and transfer request submission process. System inputs and outputs are listed below:

System Inputs

From Residents (External Entity):

- In-House Transfer Request.
- Vacancy Listing.
- Resident Apartment Vacancy Request.

From Non-Residents (External Entity):

- Applicant Information.
- New Apartment Application.
- Non-Resident Apartment Vacancy Request.

From Residential Services (External Entity):

- Approval/Rejection Acknowledgement for Vacancy Spot.
- Application Approval/Rejection Information.
- Application Creation Acknowledgement.
- Service Request Update.

System Outputs:

To Residents (External Entity):

- In-House Transfer Request Status Update (creation/ approval/ rejection).
- Vacancy Update to Residents.
- Monthly Rent Alerts; Service Request Real-time Update.
- Waitlist Information to Residents.
- Acknowledgement of Deletion of Pending Request to Residents (more than 180 days).

To Non-Residents (External Entity):

- Acknowledgement of Deletion of Pending Request to Non-Residents (more than 180 days).
- Application creation/ Update confirmation (with comments).

- Vacancy Update to Non-Residents.
- Waitlist Information to Non-Residents.
- Unique ID Creation.

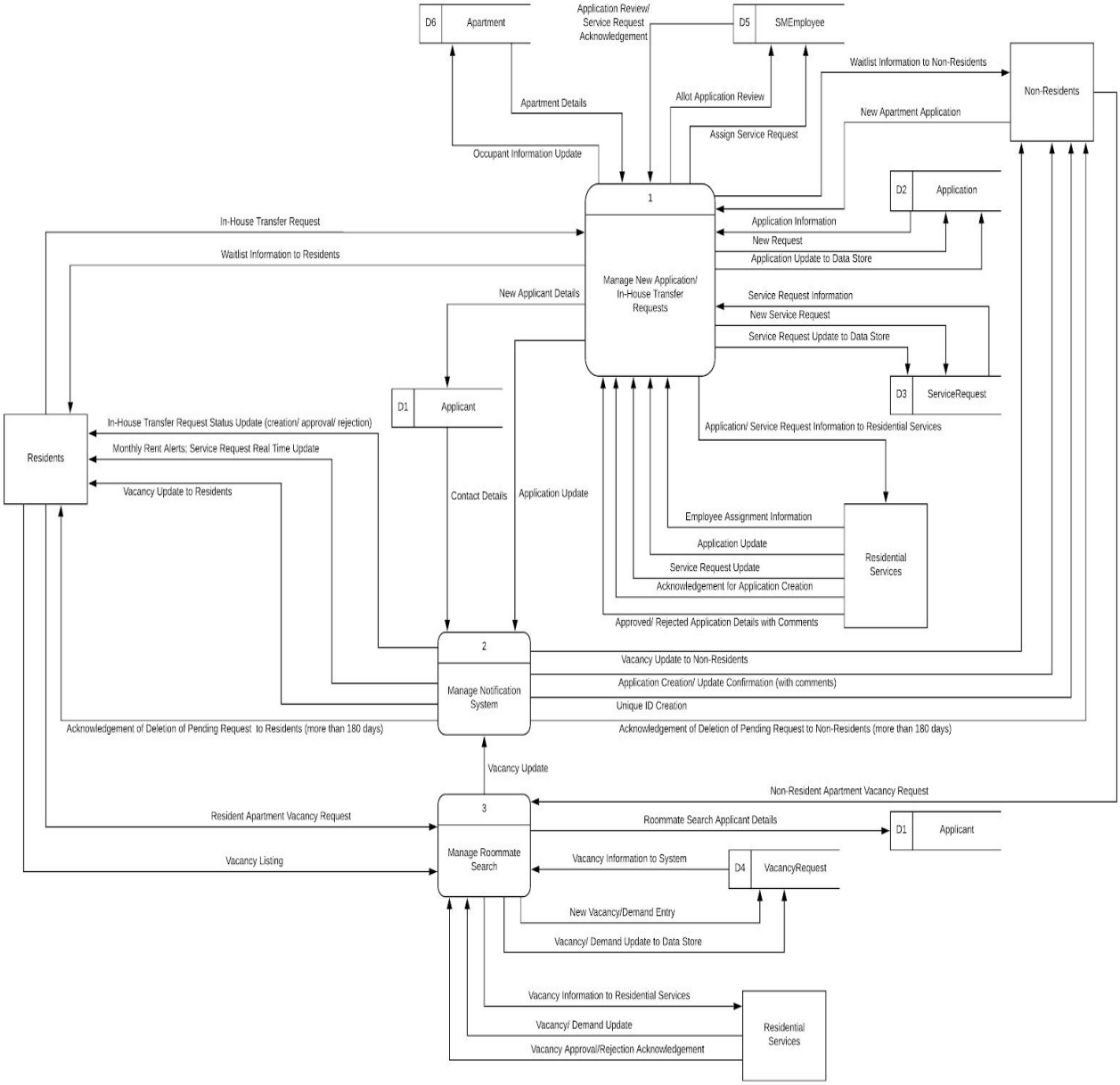
To Residential Services (External Entity):

- Non-Resident Applicant Information.
- In-House Transfer Request Application.
- Vacancy Information from Resident Listing.

2.2 Level 0 Diagram

Explanation:

The three entities described in the context diagram are retained in the Level 0 diagram. However, the system has now been expanded and has three processes and six data stores.



Data Stores:

Applicants (D1): This data store stores the information of applicants. Users can be either residents of Southern Management or non-residents. The information stored is generally name and contact information. This data is inputted to the 'Manage New Application/In-House Transfer Request System' process. This data store is being used and updated in the scope of this project.

Application (D2): This data store stores information that is relevant to applications.. This data store is also being used and updated in the scope of the project.

ServiceRequest (D3): This data store stores information about the service requests. These requests are generated by residents and assigned to Southern Management employees. This data store is also being used and updated in the scope of the project.

VacancyRequest (D4): This data store has information about the vacancy requests that are raised by applicants who could be both residents or non-residents. This data store is also being used and updated in the scope of the project.

SMEmployee (D5): This data store has information about the employees of Southern Management. The employees are assigned application reviews and service requests. This data store is also being used and updated in the scope of the project.

Apartment (D6): This data store stores information about the apartments that are listed under Southern Management. Occupant information is continually listed along with apartment information. This data store is also being used and updated in the scope of the project.

Processes:

1. **Manage New Application/In-House Transfer Request:**

   This process manages the applications submitted by residents/non-residents. It includes capturing user information and storing it in appropriate data stores. Once recorded, this information is used by other processes and entities to perform other functions. This process directly interacts with users of the system (residents and non-residents)

   Process Inputs:

   From Residents (External Entity):

   - In-House Transfer Request.

   From Non-Residents (External Entity):

   - New Apartment Application.

From Residential Services (External Entity):

- Employee Assignment Information.
- Application Update.
- Acknowledgement for Application Creation.
- Service Request Update.
- Approved/ Rejected Application Details with Comments.

From ServiceRequest (Data Store D3):

- Service Request Information.

From SMEmployee (Data Store D5):

- Application Review/ Service Request Acknowledgement.

From Apartment (Data Store D6):

- Apartment Details.

Process Outputs:

To Applicant (Data Store D1):

- New Applicant Details.

To ServiceRequest (Data Store D3):

- New Service Request.
- Service Request Update to Data Store.

To SMEmployee (Data Store D5):

- Assign Service Request.

To Apartment (Data Store D6):

- Occupant Information Update.

To Manage Notification System (Process):

- Application Update.

To Residents (External Entity):

- Waitlist Information to Residents.

To Non-Residents (External Entity):

- Waitlist Information to Non-Residents.

To Residential Services (External Entity):

- Application/ Service Request Information to Residential Services.

2. **Manage Notification System:**

This process manages all alerts and notifications of the new system. This includes receiving updates from the 'Residential Services' and the 'Applicant' data store and sending relevant notifications to external entities mainly residents and non-residents. This process interacts with the residents and non-residents.

Process Inputs:

From Applicant (Data Store D1):

- Contact Details.

From Search Roommates (Process):

- Vacancy Update.

From Manage New Application/In-House Transfer Request (Process):

- Application Update.

Process Outputs:

To Residents (External Entity):

- In-house Transfer Request Status Update (creation/ approval/ rejection).
- Monthly Rent Alerts; Service Request Real Time Update.
- Vacancy Update to Residents.
- Acknowledgement of Deletion of Pending Request to Residents (more than 180 days).

To Non-Residents (External Entity):

- Vacancy Update to Non-Residents.
- Application Creation/ Update Confirmation (with comments).
- Unique ID Creation.

- Acknowledgement of Deletion of Pending Request to Non-Residents (more than 180 days).

3. **Manage Roommate Search:**

This process manages vacancy listings for residents as well as non-residents. It fetches applications from both these entities and updates the 'VacancyRequest' data store accordingly with new requests and listings. This information is used by the 'Residential Services' which sends acknowledgement of approval or rejection back to the process for dissemination to residents and non-residents through the notification system.

Process Inputs:

From Residents (External Entity):

- Vacancy Listing.
- Resident Apartment Vacancy Request.

From Non-Residents (External Entity):

- Non-Resident Apartment Vacancy Request.

From Residential Services (External Entity):

- Vacancy Approval/Rejection Acknowledgement.
- Vacancy/Demand Update.

From VacancyRequest (Data Store D4):

- Vacancy Information to System

Process Outputs:

To Manage Notification System (Process):

- Vacancy Update.

To Applicant (Data Store D1):

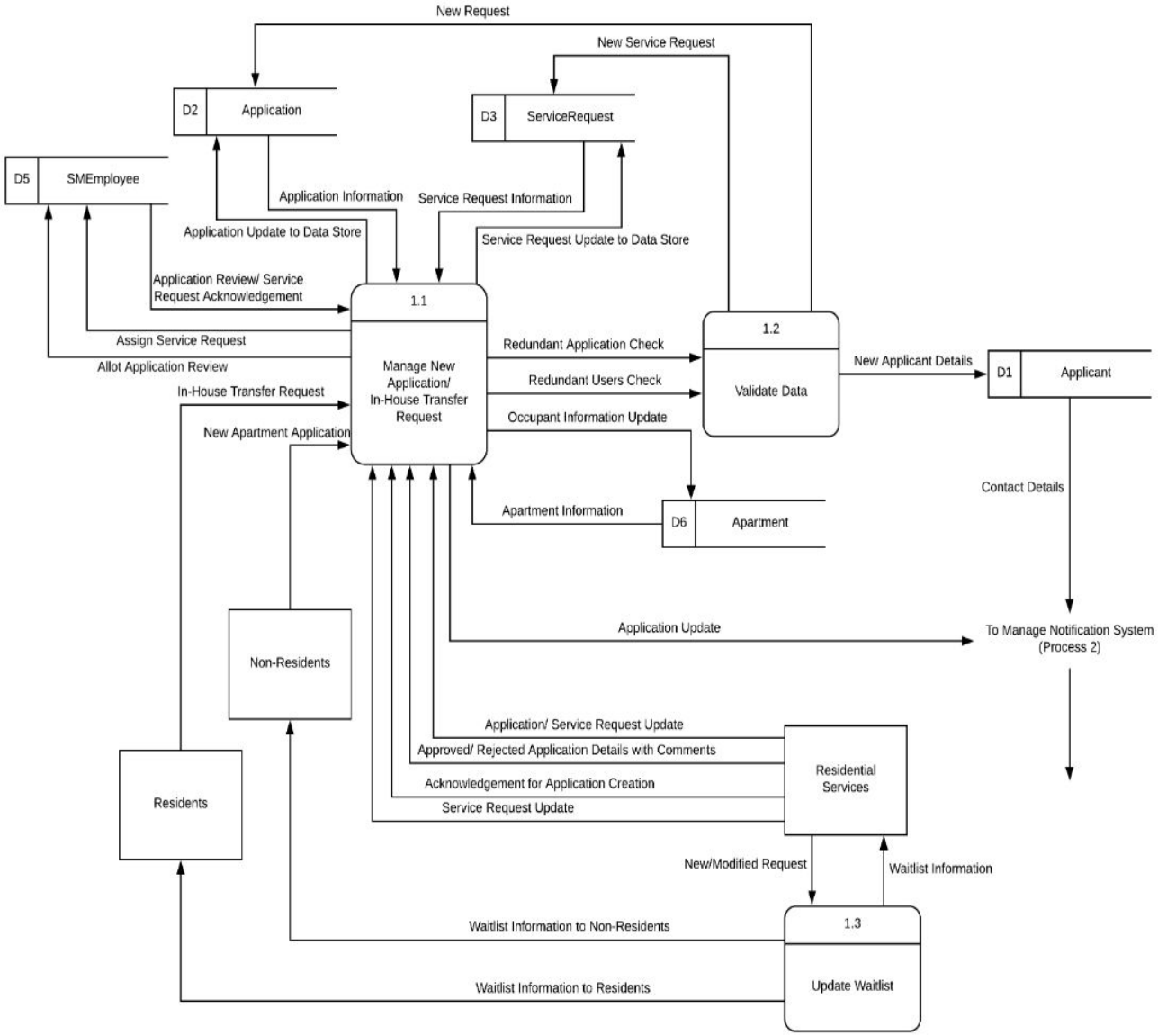
- Roommate Search Applicant Details.

To VacancyRequest (Data Store D4):

- New Vacancy/Demand Entry.
- Vacancy/Demand Update to Data Store.

2.3 Level 1 Diagram

2.3.1 Level 1 : Process 1



Explanation:

The Level 0 diagram has now been split into three Level 1 diagrams. Each process at Level 0 has been explained separately. The ‘Level 1: Process 1’ explains the process ‘Manage New Application/In-house Transfer Request’. This diagram shows management of all the new applications and in-house transfer requests through three new sub-processes and the existing data stores.

Processes:

1.1 Manage New Application/In-House Transfer Request:

This process gets application requests from residents and non-residents. It passes on these applications to the validation process which performs redundancy checks on these applications. It receives acknowledgements based on approvals/ rejections and service request updates from 'Residential Services' and later on, sends application updates to the notification system.

Process Inputs:

From Residents (External Entity):

- In-house Transfer Requests.

From Non-Residents (External Entity):

- New Apartment Application.

From Residential Services (External Entity):

- Approved/Rejected Application Details with Comments.
- Application/ Service Request Update.
- Acknowledgement for Application Creation.
- Service Request Update.

From SMEmployee (Data Store D5):

- Application Review/ Service Request Acknowledgement.

From Apartment (Data Store D6):

- Apartment Information.

From Application (Data Store D2):

- Application Information.

From ServiceRequest (Data Store D3):

- Service Request Information.

Process Outputs:

To Validate Data (Process 1.2):

- Redundant Applications Check.

- Redundant Users Check.

To Apartment (Data Store D6):

- Occupant Information Update.

To Manage Notification System (Process 2):

- Application Update.

To SMEmployee (Data Store D5):

- Allot Application Review.
- Assign Service Request.

To Application (Data Store D2):

- Application Update to Data Store.

To ServiceRequest (Data Store D3):

- Service Request Update to Data Store.

1.2 Validate Data

This process takes inputs from the process 'Manage New Application/In-House Transfer Request' and performs validation checks on users and applications to check for redundancy. This helps manage the issue of duplicate applications in the system. Using these inputs, this process sends new applicant details to the 'Applicant' data store (D1), 'Application' data store (D2) and 'ServiceRequest' data store (D3).

Process Inputs:

From Manage New Application/ In- House Transfer Request (Process 1.1):

- Redundant Applications Check.
- Redundant User Check.

Process Outputs:

To Applicant (Data Store D1):

- New Applicant Details.

To Application (Data Store D2):

- New Request.

To ServiceRequest(Data Store D3):

- New Service Request.

1.3 Update Waitlist

This process takes in new or modified requests from 'Residential Services' and updates the waitlist accordingly. Later, it sends this information to all entities involved, namely, 'Residents', 'Non-Residents', 'Residential Services'. This waitlist information is continuously updated in the 'System Database' data store(D2).

Process Inputs:

From Residential Services (External Entity):

- New/ Modified Request.

Process Outputs:

To Residents (External Entity):

- Waitlist Information to Residents.

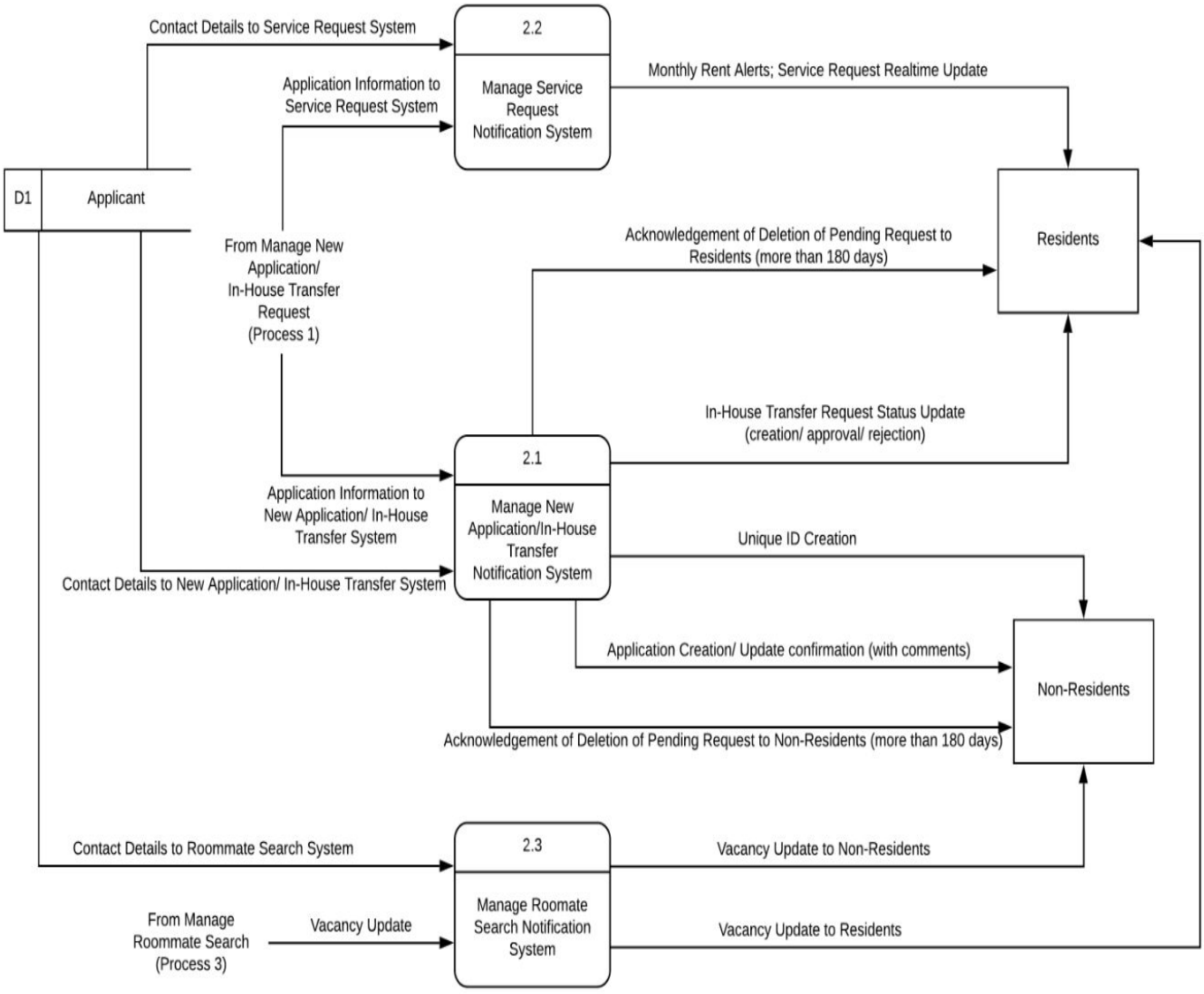
To Non-Residents (External Entity):

- Waitlist Information to Non-Residents.

To Residential Services (External Entity):

- Waitlist Information.

2.3.2 Level 1 : Process 2



Explanation:

The 'Level 1: Process 2' diagram explains the notification system flow. The 'Manage Notification System' process has been further divided into three sub-processes in this diagram. These processes receive information from Processes 1 and 3 during the scope of the project at different points. This process decides which notifications will go to which entity (user) and how it will flow through the system.

Processes:

2.1 Manage New Application/In-House Transfer Notification System

This process gathers new applications and in-house transfer requests from 'Process 1' along with contact details of the users from data store D1 (Applicant). The primary purpose of this process is to send out acknowledgments to residents and non-residents regarding their application status.

Process Inputs:

From Manage New Application/ In-House Transfer Request (Process 1):

- Application Information to New Application/ In-House Transfer System.

From Applicant (Data Store D1):

- Contact Details to New Application/ In-House Transfer System.

Process Outputs:

To Residents (External Entity):

- In-House Transfer Request Status Update (creation/approval/rejection).
- Acknowledgement of Deletion of Pending Request to Residents (more than 180 days).

To Non-Residents (External Entity):

- Application Creation/ Update confirmation (with comments).
- Acknowledgement of Deletion of Pending Request to Non-Residents (more than 180 days).
- Unique ID Creation.

2.2 Manage Service Request Notification System

This process has been created for the purpose of sending rent alerts and service request updates to residents of Southern Management. It takes its inputs from 'Process 1' and contact details from the data store 'Applicant' (D1).

Process Inputs:

From Applicant (Data Store D1):

- Contact Details to Service Request System.

From Manage New Application/ In-House Transfer Request (Process 1):

- Application Information to Service Request System.

Process Outputs:

To Residents (External Entity):

- Monthly Rent Alerts; Service Requests Real-time Update.

2.3 Manage Roommate Search Notification System

This process solely focuses on sending out updates for vacancy requests that are created by residents of Southern Management and non-residents. It takes in the contact information from the 'Applicant' data store (D1) and 'Process 3' and sends out vacancy updates to residents and non-residents.

Process Inputs:

From Manage Roommate Search (Process 3):

- Vacancy Update.

From Applicants (Data Store D1):

- Contact Details to Roommate Search System.

Process Outputs:

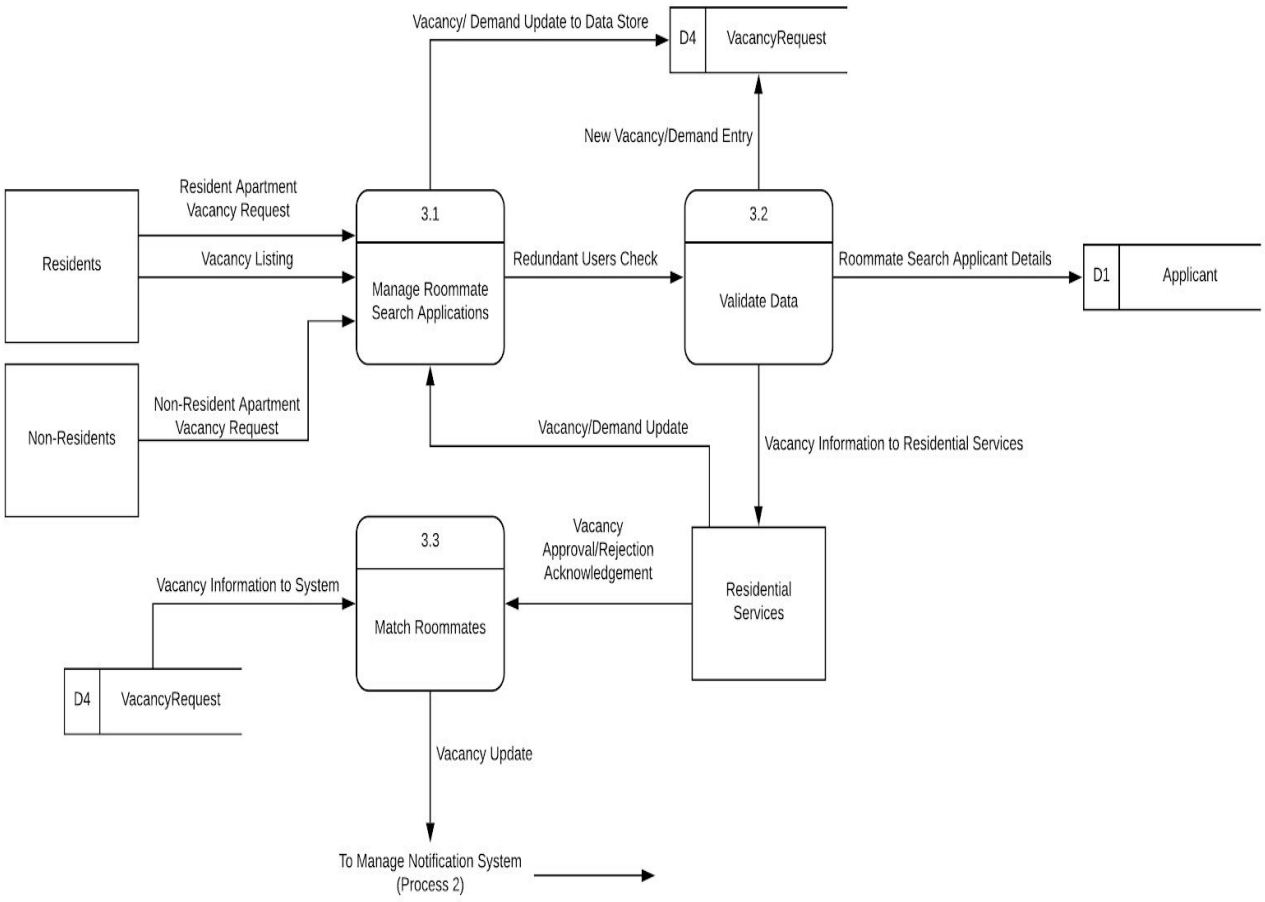
To Residents (External Entity):

- Vacancy Update to Residents.

To Non-Residents (External Entity):

- Vacancy Update to Non-residents.

2.3.3 Level 1 : Process 3



Explanation:

The 'Level 1: Process 3' diagram focuses on the management of the roommate search functionality that is being used by both residents and non-residents. 'Process 3' has been divided into three sub-processes to explain the overall flow. This process updates and uses the data stores through the scope of the project.

Processes:

3.1 Manage Roommate Search Applications

This process takes as inputs all the vacancy requests of residents and non-residents. In addition to that, it also fetches vacancy listings from residents of Southern Management. This information is sent to the validation process for redundancy checks.

Process Inputs:

From Residents (External Entity):

- Vacancy Listings.
- Resident Apartment Vacancy Request.

From Non-Residents (External Entity):

- Non-Resident Apartment Vacancy Request.

From Residential Services (External Entity):

- Vacancy/Demand Update.

Process Outputs:

To Validate Data (Process 3.2):

- Redundant Users Check.

To VacancyRequest (Data Store D4):

- Vacancy/ Demand Update to Data Store.

3.2 Validate Data

This process validates the information coming in from 'Process 3.1' and updates the data stores accordingly. It stores the new applicant details in the 'Applicant' data store (D1) and the new vacancy or demand entry into 'VacancyRequest' data store (D4). This information is later passed on to the 'Residential Services'.

Process Inputs:

From Manage Roommate Search Applications (Process 3.1):

- Redundant Users Check.

Process Outputs:

To Applicant (Data Store D1):

- New Applicant Details.

To VacancyRequest (Data Store D4):

- New Vacancy/ Demand Entry.

To Residential Services (External Entity):

- Vacancy Information.

3.3 Match Roommates

This process has been created to send vacancy updates to the notification system. These notifications are sent by the 'Residential Services' after gathering this information from the VacancyRequest (data store D4).

Process Inputs:

From Residential Services (External Entity):

- Vacancy Approval/ Rejection Acknowledgement.

From VacancyRequest (Data Store D4):

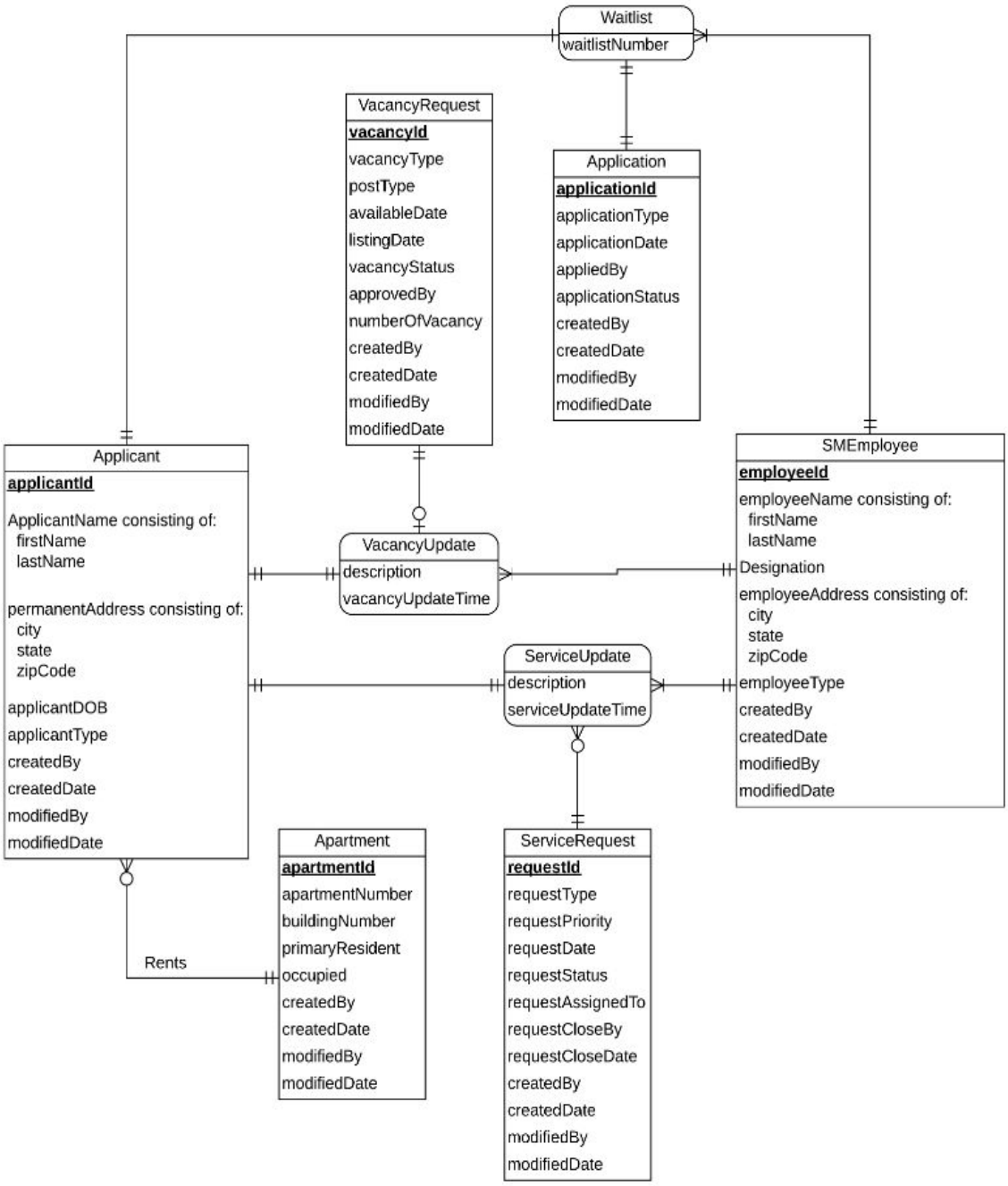
- Vacancy Information to System.

Process Outputs:

To Manage Notification System (Process 2):

- Vacancy Update.

3. Proposed ER Diagram



Explanation:

The proposed ER Diagram helps us identify the different system elements and their relationships with each other. We have a total of 9 entities including 3 associative entities. They are listed and explained as follows:

1. **Applicant**: This entity holds the information of residents and non-resident applicants. It stores general information about the applicants including contact information.

2. Apartment: This entity holds the information about the apartments in the Southern Management communities. It also holds information about primary occupants.

3. SMEmployee: This entity stores information about the Southern Management employees. It holds details such as their basic information and type of employment.

4. ServiceRequest: This entity stores information about the service requests that are initiated by residents of the community. It includes basic information like what type of service request it is and what its priority is.

5. Application: This entity stores information about applications. This includes new applications and in-house transfer requests. It also holds the creation date and type of the application.

6. VacancyRequest: This entity stores information about vacancies in general. This includes applications from residents and non-residents.

7. Waitlist (Associative Entity): This entity is a relational entity between Applicant, Application and SMEmployee. It holds the waitlist information.

8. VacancyUpdate (Associative Entity): It is a relational entity between VacancyRequest, Applicant and SMEmployee. This entity provides updates about vacancies.

9. ServiceUpdate (Associative Entity): This is a relational entity between Applicant, ServiceRequest and SMEmployee. This entity provides updates on service requests.

Relations:

We have a total of three ternary relationships and one binary relationship. They are listed as follows:

1. Rents (Binary):

    One Applicant to exactly one Apartment

    One Apartment to zero or many Applicant.

2. Waitlist (Ternary):

One Applicant, one Application to one or many SMEmployee

One Application, one SMEmployee to exactly one Applicant.

One Applicant, one SMEmployee to exactly one Application

3. VacancyUpdate (Ternary):

One Applicant, one VacancyRequest to one or many SMEmployee

One VacancyRequest, one SMEmployee to exactly one Applicant.

One Applicant, one SMEmployee to zero or one VacancyRequest

3. ServiceUpdate (Ternary):

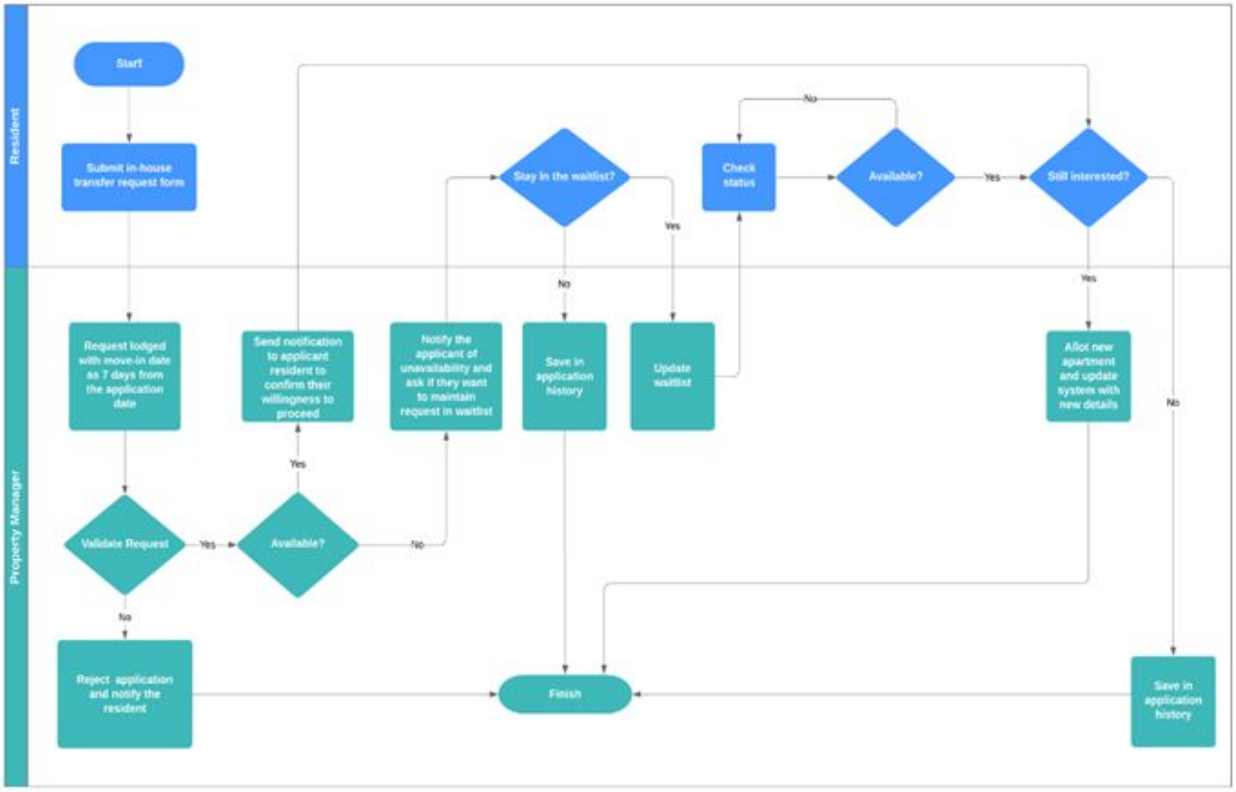
One Applicant, one ServiceRequest to one or many SMEmployee

One ServiceRequest, one SMEmployee to exactly one Applicant.

One Applicant, one SMEmployee to zero or many ServiceRequest.

4. Process Flow Diagram

<u>Function</u>: Provision for online in-house transfer request



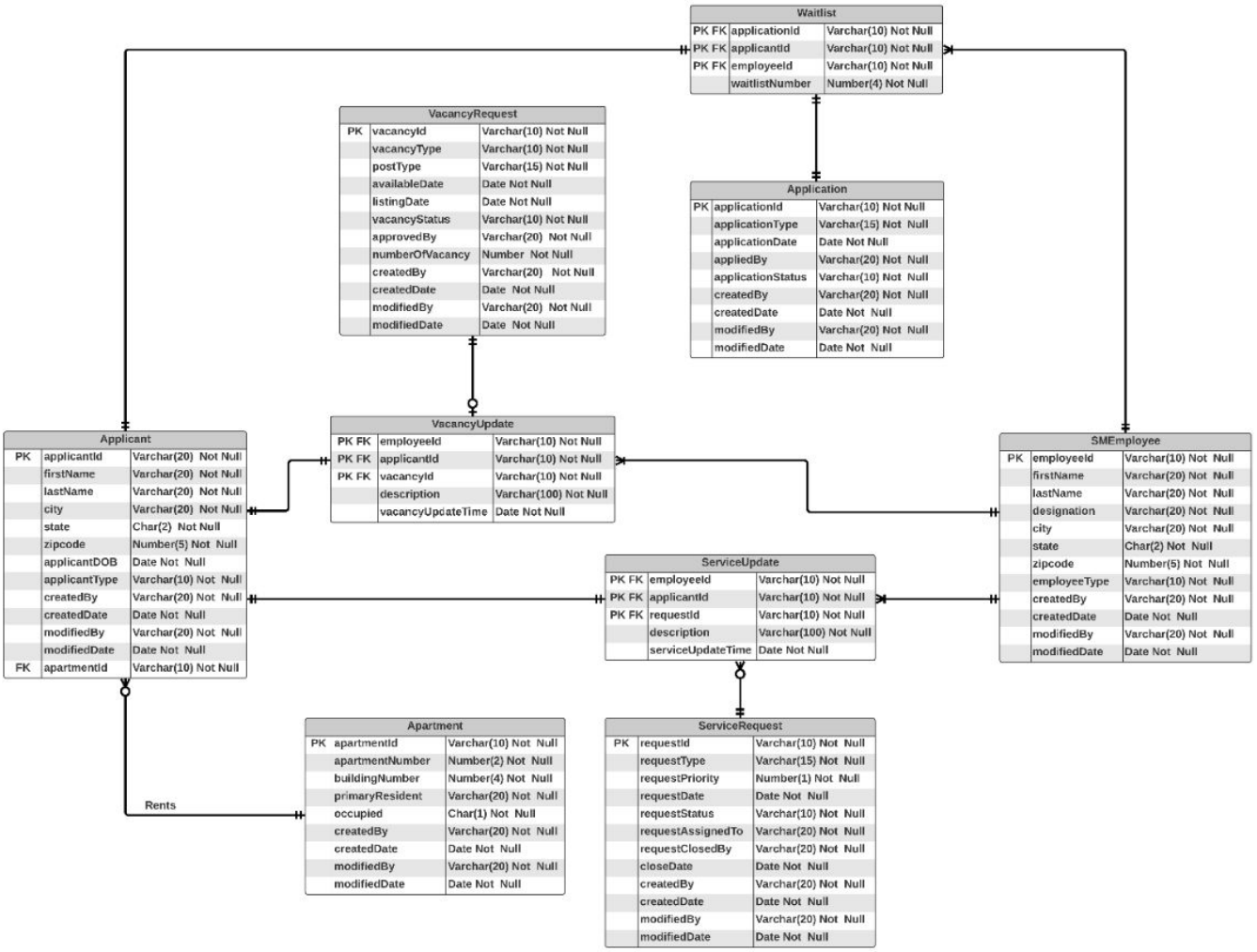
5. Conclusion

This concludes the first 2 phases – the Planning and the Analysis phase of the project. During the planning phase, the scope and various feasibilities of the project were discussed. A consolidated project plan was put into place. Based on this, the analysis phase was initiated, and a comprehensive set of analyses were laid out.

The data flow diagrams have been worked well upon and are granular to a great level and intended to help in the following phases, like developing an accurate design plan, for starters. The overview created using the context diagrams helped us draft the Level 0 and Level 1 diagrams. Collectively with the help of the ER diagram and Process Flow diagram, we can confidently infer that our recommendations to enhance the Southern Management system are up to the mark. This will improve the efficiency of their system and improve user experience and benefit them significantly in the long run and the investment will significantly be outweighed by the rewards it will bring.

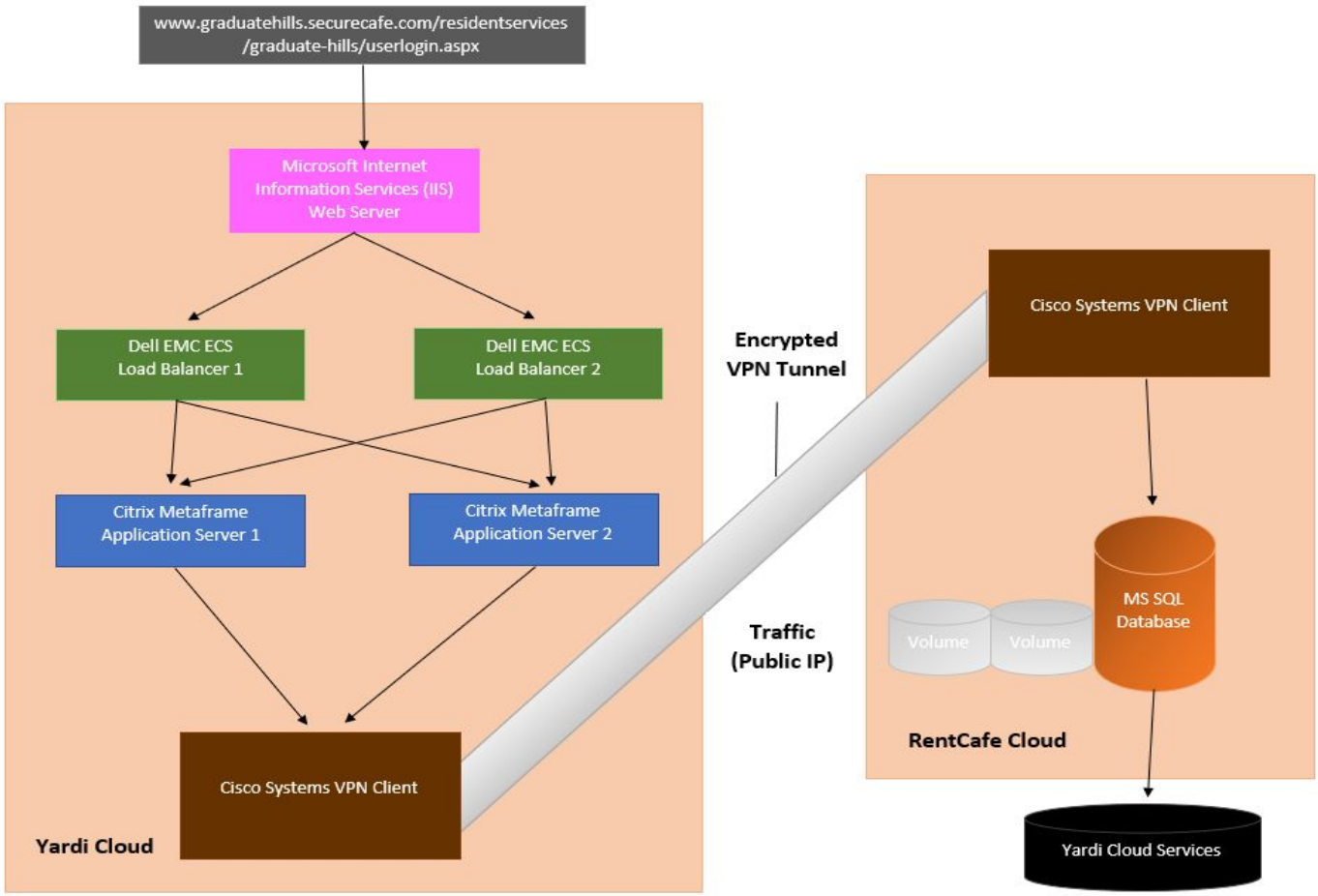
PHYSICAL ERD AND DESIGN ARCHITECTURE

PHYSICAL ERD



 # Note: In case the cardinality of relations are not clear in Physical ERD, please refer to the Logical ERD as the relations are the same.

DESIGN ARCHITECTURE



Our project is an enhancement project and thus we used a hybrid architecture to design cloud architecture for our system. Our plan is to design the application on cloud services provided by Yardi Systems, which is also used by RentCafe application for Southern Management.

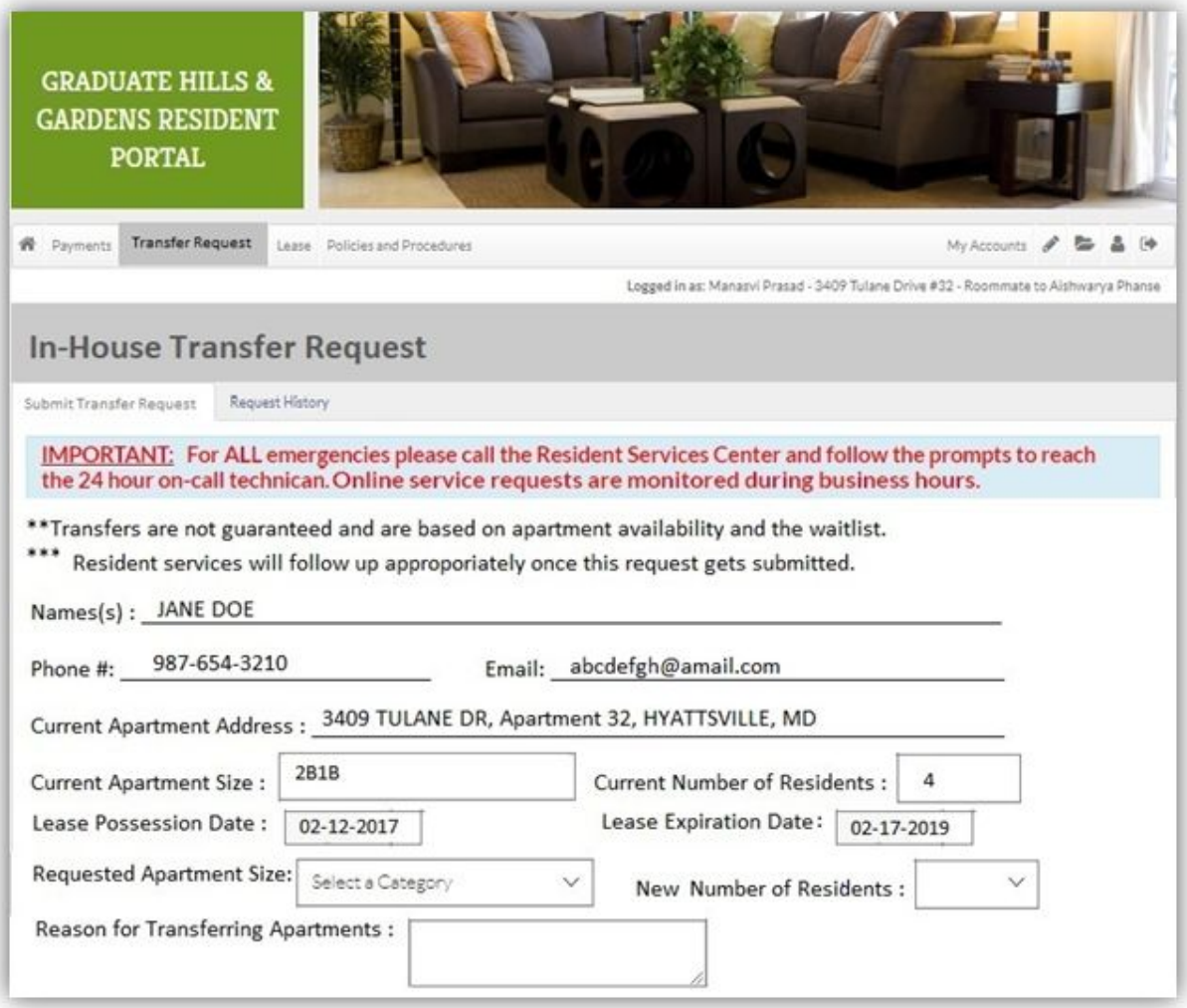
- When a user enters the Southern Management website in URL, the DNS would look up the IP address and redirect the user to a web server provided by Microsoft Internet Information Services (IIS) to the Southern Management application.
- The requests are sent to application servers through load balancers. The application uses Dell EMC ECS load balancers. It's purpose is to evenly distribute requests to servers to avoid traffic on a particular server.
- The application servers are provided by Citrix Metaframe. There are 2 application servers in our system which process the requests.
- The processed requests and responses are transferred to and from the RentCafe application through Cisco System VPN Client. These VPN managers are encrypted for high security.
- The processed requests and responses are stored in MS SQL database servers on Yardi's cloud.

IMPLEMENTATION DETAILS

1. Development Details

1.1 User Interface Details

1.1.1 Function 1 UI: Online In-House Transfer Request



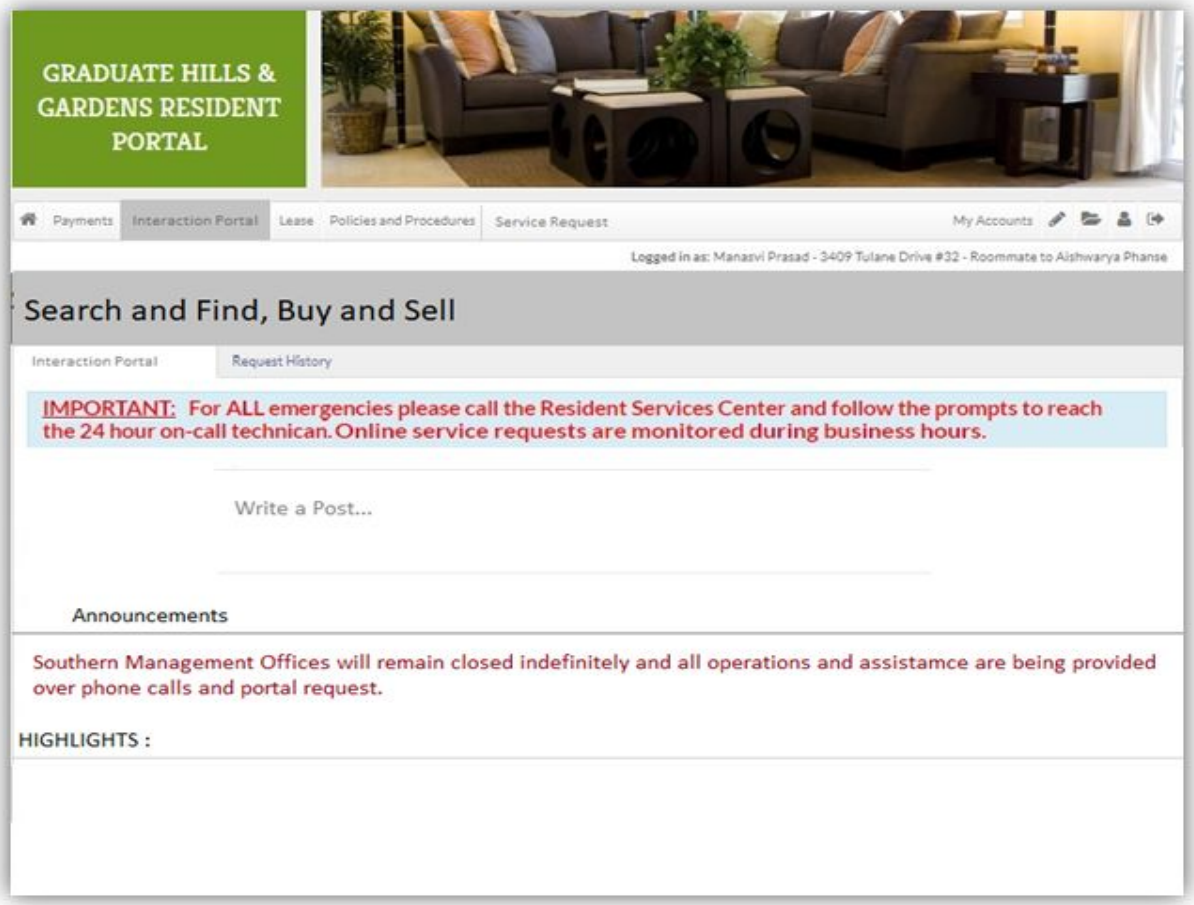
The online in-house transfer request system will contain an online form on the Southern Management Portal as shown above. The features of the online form are mentioned below:

1. The account details (name, phone number, email, current address, apartment size, number of occupants and lease dates) would be automatically populated in the form.

2. The user would be asked to enter the reasons for the request (text field), number of new residents (dropdown menu with range 1-4), and requested apartment size (dropdown menu with 5 apartment categories).
3. The form clearly mentions the conditions of request processing and follow-up by Southern Management Personnel.

The form layout has been presented in clear and intuitive format requiring minimal user input and capturing the required information to process requests.

1.1.2 Function 2 UI: Search and Find New Roommates

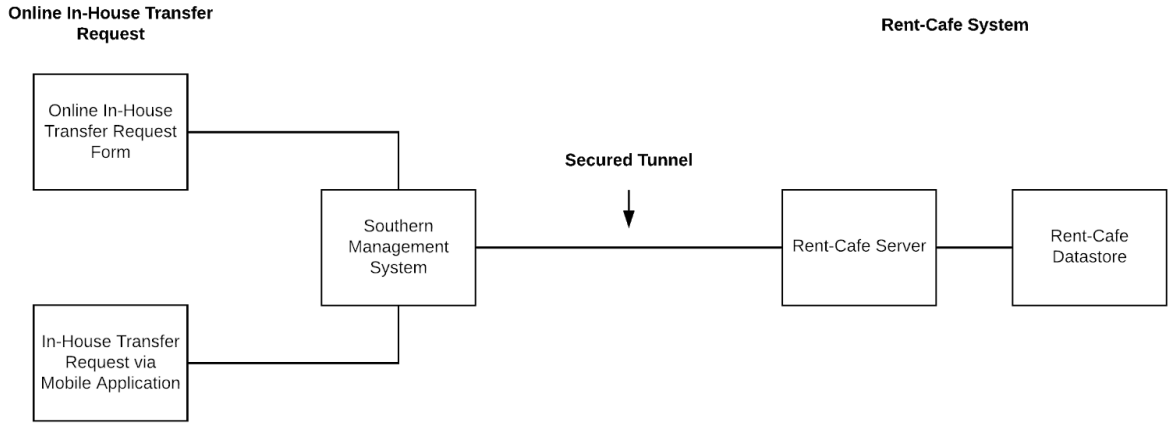


The online system to search and look for new flatmates provides an user-friendly interface where the user is provided with the text field to enter their request. The Highlights section provides users a scrollable interface to view postings by other residents.

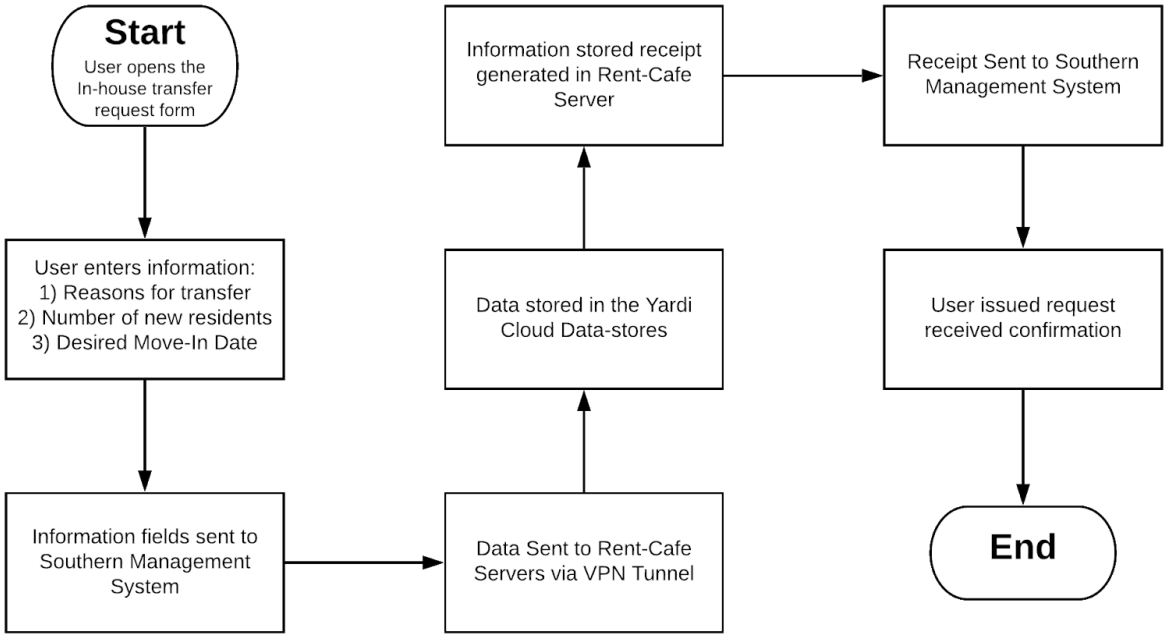
The whole UI interface provides an easy-to-interact platform for residents to connect and look for potential roommates.

1.2 Integration

Through the integration of the online in-house transfer request system into the Rent-Cafe system, the handling of in-house transfer requests via electronic forms should make it convenient for Southern Management personnel to process the requests. The goal of the integrated system is to provide bi-directional information electronically without using paper documentation. This will be achieved by a user-friendly online form to enter information and properly inform Southern Management and Rent-Cafe to process and use the incoming information. The below shows the equipment setup for the systems integration plan:



The information flow for the new integration system is showcased below:



2. Documentation Details

2.1 System Documentation:

Function 1: Online in-house transfer request

Infrastructure Setup:
Based on the details of input from the user in the form, there will be validations of UI at the Front-end on the RentCafe server. Also, there would be Southern Management business rules validation regarding Lease information, Inspection rules, Willingness to sign a 12 month lease and so on at the backend.

Transfer guideline: The document for Transfer Officer to refer to when handling an incoming in-house transfer request form, about Data processing and details to add for ''Office Use'' like Enrollment Verification, Payment history, Date of Apartment Inspection, etc.

Function 2: Search and find potential roommates

Infrastructure Setup:
Based on the business rules of community guidelines, initiate at backend to raise a notification to Administrator about a violation at the front-end for action.
Also, there would be Southern Management business rules validation regarding Lease information, Inspection rules, Willingness to sign a 12-month lease and so on at the backend.

Function 3: A transparent waitlist

Infrastructure Setup:
Based on the date of application, date of move-in and apartment type, validate the details and update the waitlist on the backend using the user input.

Function 4: Customer notification system to provide rent alerts, service request updates etc.

Infrastructure Setup:
Based on any status change of service request, update the backend to send across a notification email and SMS and ask for review. Every second and fourth day of the month, run a job at the backend to send out notification for Rent reminders.

2.2 Online documentation:

Knowledge based articles:

Guidelines Document: Community guidelines pertaining to Business rules to refer to for interaction between members on the portal.

FAQ document: About user details of in-house transfer form and procedure of in-house transfer and its payment guidelines.

3. Infrastructure Setup

Details of load requirements:

2000 concurrent users between 10 am to 8:00 p.m, volume drops to 600 users. About 3000 users in the first week of the month.

Details of vendor interactions:

The system will interact with the Telecom Partner for SMS notification services.
The system uses RentCafe server for user inputs for portals, requests and payments.

TEST PLANS (TEST DOCUMENTS)

Included as a separate file.

Test cases run: 20

Types of testing done:

- Unit Testing
- Performance Testing
- Stress Testing
- Regression Testing

IMPLEMENTATION PLAN

Included as a separate file.

LESSONS LEARNED

- **Communication Management:** Weekly updates on the status to stakeholders would have reduced the enquiry management hours.
- **Time Management:** Buffer time should have been retained in the schedule for troubleshooting when problems arise.
- **Testing Technique:** Negative testing should have been included to check if the system can handle unwanted input or unusual user behavior.
- A majority time should have been given to the training phase of the project. Giving this a priority during the planning phase would have reduced the extension of the deadline of the project.
- Changes in the plan are inevitable in any project. Keeping documents of the mistakes and fixes handy can help in future iterations in the same project as well as different projects of the company in the future.
- Change Management should be given equal importance along with the technical SMEs for an implementation project, such as this, for full value optimization.