HomePairs:
Software Requirements Specification
Version 3.0

Toopairs
Computer Science Department
California Polytechnic State University
San Luis Obispo, CA USA

December 1, 2019

Contents

Credits

Name	Date	Role	Version
Adam Berard	October 10, 2019	Author of 2.4 and Revisor and Co-Author of 4	1.0
Tommy Bergmann	October 10, 2019	Author of 1.1 and 1.4	1.0
Cesar Chácon	October 10, 2019	Co-Author of Section 2 and Appendix A	1.0
Eeron Grant	October 10, 2019	Author of Section 2.6 and 2.8, Co-Author of Horizontal Prototype	1.0
Sarah MacDougall	October 10, 2019	Lead Author of Horizontal Prototype, Jira Manager	1.0
Adam Berard	October 23, 2019	Section 4, Monica Persona	2.0
Cesar Chacon	October 23, 2019	General Peer Reviewer	2.0
Cesar Chacon	December 3, 2019	General Reviewer	3.0

Revision History

Name	Date	Reason for Changes	Version
Adam Berard	October 10, 2019	Revised certain items from 4.1.3.	1.0
Tommy Bergmann	October 10, 2019	Thematic changes to 2.3	1.0
Cesar Chacon	October 10, 2019	Section 2.1, 2.2, 2.5	1.0
Eeron Grant	October 10, 2019	Reviewed Section 3	1.0
Sarah MacDougall	October 10, 2019	Section 3.5, 6	1.0
Adam Berard	October 13, 2019	Revised 2.5 and 2.7	2.0
Cesar Chacon	October 20, 2019S	Revised Section 4 and Section 1	2.0
Eeron Grant	October 16, 2019	Updated Section 3 Section 2.7	2.0
Sarah MacDougall	October 17, 2019	Revised Section 2.7	2.0
Eeron Grant	October 19, 2019	Added Sections 3.6,3.7 and 3.8	2.0
Cesar Chacon	December 3, 2019	Updated document to reflect new defintion of Property Managers	3.0

1 Introduction

1.1 Purpose

The SRS contains the limitations and requirements for the Roopairs-PolyRents application 'HomePairs', covering all functional and nonfunctional features that are expected in version 3.0. This document will be used as a reference by the development team for what product they are expected to develop.

1.2 Intended Audience

The SRS is intended for the CEO's and CTO's of Roopairs and PolyRents.

1.3 Project Scope

The primary purpose of this application is to connect property managers and tenants to service providers to make the process of fixing appliances and other general repairs both quick and easy. This application is being developed directly with Roopairs to connect their service providers with property managers who need repair service for their property. Although we are not working directly with PolyRents, they are our target source of demand in the housing market.

1.4 References

Roopairs Proposal: Click Here

2 Overall Description

2.1 Product Perspective

The product being developed is a stand-alone software application that will use Roopairs' Service Provider Network API to address the issue of property manager and tenant communication based on service job requests. This application will be an extension of the PolyRents website which mainly addresses the issue of finding housing for tenants and managing housing applications for property manager. Our application will be made with React Native in order to achieve seamless partnership integration with the PolyRents' website front end which is built with React.

2.2 Product Features

The main features of our application will focus on two parties: tenants and property manager.

1. Tenants will be able to send service requests to their property manager.
2. Property manager and tenants can view past and current jobs.
3. Property managers can approve and send job service requests to the Roopairs service provider network.
4. Both property manager and tenants will receive notifications when a Job or Job Request status changes.
5. Property managers can view reports and analysis on Jobs.
6. Property managers and tenants will have a means of communication.

2.3 User Personas

Name:	Monica Johnson
Age:	47
Occupation:	Manager of the local botanical garden
Years Managing Property:	No Prior Experience
Hometown:	Santa Monica, CA
Character:	Persistent and Driven
Bio:	Monica Johnson (born Monica Moretti) is a hard working mom of two beautiful young daughters. She started working at the botanical garden right out of high school and has worked her way up to manager. Her husband is an environmental engineer and between their two salaries they do quite well. Monica is an only child and with the recent death of her two parents in a heli-tour accident in the Bahamas, she was the sole inheritor of their house located about 30 minutes away from where her and her husband are currently living. She would like to set the property up as a rental so the family can have a steady stream of income for the rest of time.
Goals/Needs:	Goal: Rent out her newly inherited house. Needs: Help figuring out how to go about managing a property since both her and her husband are busy and have no experience with managing properties.

Frustrations/Fears/Challenges:	Monica is worried that they will be in over their heads with this project and is concerned that it might be expensive to get the house in renting shape. Many of the appliances are older and she knows they will need to be serviced or replaced within ten years. A big challenge for Monica will be finding the time to get to the property since it is 30 minutes away and she already has a lot on her hands between the kids and her work.
Preferred Channels:	Calling, Text Messaging, E-Mail
Motivations:	Monica knows that if she can get the house rented then her family will have a steady source of income for years to come and eventually even her little girls could have the same source of income. Her family is the most important thing in her life and she wants to secure the best possible future for them.
How can we help?	We can help quell some of Monica's fears about property management by giving her a simple and easy to manage application that is accessible on her phone that will assist in appliance repairs for her property. One of her worries was the old appliances at the house so this should be a big help for her.
Tech Acumen	Fluent with Technology
Created By:	Adam Berard

Name:	Property Manager "Uncle" Carl Mahoe
Age:	51
Occupation:	Warehouse worker at Trader Joes
Years Managing Property:	3.5 years
Hometown:	Monterey,CA
Character:	Rugged, Comedic, and Casual
Bio:	Carl's cousin, Jim Wallet, grew up in Monterey. Jim had acquired 5 houses and 2 small apartment blocks in the area. But Jim is already employed as an executive at Capital Insurance. So Carl lives a small, rundown house by the beach, and rent is free in return for acting as a Property Manager for Jim's properties.

Goals/Needs:	Goal: trip to the Bahamas Goal: restore a Volkswagen bus or old motorhome as a fallback for retirement Needs: more time in his day, a way to minimize the time it takes to address tenant complaints and repair requests.
Frustrations/Fears/Challenges:	Any costs for repairs eventually come out of Jim's pocket, but Carl knows he's expected to keep costs to a minimum and justify all repairs. He struggles to keep track of too many different jobs, so he prefers to work with the same one or two plumbers for all 7 properties. He keeps a bunch of hand-written and printed receipts in a few manila folders on his desk. Sometimes it takes him 20 minutes to find a certain paper, but he knows it's in one of those folders. He'd be furious if he lost a receipt and had to pay for something out of pocket. Doesn't know when repair requests will come in, so he doesn't feel like he can take long trips anywhere. Nowadays, he's never gone for more than 3 days at a time.
Preferred Channels:	Cell Phone (Strongly prefers calling over text), Home Phone, Yahoo Email
Motivations:	Gets to live by the beach rent free. Has a pretty easy yet stable living situation
How can we help?	We can help Carl manage repair requests quickly, and when he wants to take trips, he'll be able to manage some of his responsibilities remotely. We also try to help Carl better organize his contacts and future messages, and even allow him to upload photos of physical documents if he prefers to keep his current system.
Tech Acumen	Reluctant but not clueless. Capable user
Created By:	Sarah MacDougall

Name:	Tenant Anzaldúa Miyamoto
Age:	34
Occupation:	Desk Employee at Wells Fargo
Hometown:	Rio de Janeiro, Brazil
Character:	Nervous, Polite

Bio:	Mrs. Miyamoto is a Brazillian immigrant who has lived in several different countries throughout her life. She grew up with a Japanese father and Chilean mother adopting both languages from her parents, however, her English is very limited. She is currently in America with a greencard hoping to gain her citizenship. She currently is renting a home in El Paso, Texas with her Haitian husband Dominique Fluer. She is raising 3 children, 2 of whom are her biological children and the remaining is the child of her husband's deceased brother.
Goals/Needs:	1. Goal: To gain U.S. citizenship. 2. Goal: Live in California. 3. Goal: Support family in Brazil after financial situation has stabilized. 4. Need: Full recovery of security deposit after the time of her lease has ended.
Frustrations/Fears/Challenges:	1. She fears of losing a significant amount of her Security Deposit from damages. 2. She currently struggles speaking English to most people in her area. 3. She is raising 3 children whom are all under the age of 11 years. 4. Her property's Property Manager is often hostile with their family.
Preferred Channels:	Cell Phone, House Phone, Email
Motivations:	1. Gain citizenship so she can help her family back in Brazil. 2. Supporting her husband to the best of her ability. 3. Granting her children a chance at a stable life. 4. Achieve the American Dream.

How can we help?	We can help by providing her the resource to easily report damages and request services to her current home. Since she is consistently occupied with her work and family, she finds it difficult to report these to her home's property manager. It is also pretty difficult for her to communicate with the property manager due to the language barrier so she resolves this issue with her oldest child of 10 years. Providing Mrs. Miyamoto with a platform to request these services can severely reduce the issues of the language barrier and can allow her to report and request services in a very efficient and convenient manner.
Tech Acumen	Little to none. Her kids do everything for her.
Created By:	Eeron Grant

Name:	Tenant Frat Boy Fred
Age:	21
Occupation:	Third year Business student
Years Renting:	3 years
Hometown:	San Diego,CA
Character:	Rowdy
Bio:	Fred is a third year at the University of Arizona who has been a part of his fraternity since his first semester at the school. He has lived in the house that him and his frat brothers rent since his sophomore year. While the house isn't the fraternity's main house, parties still get thrown at the house and there are people coming through the house all the time. As a result, the house is rarely clean and the appliance all get heavily used (as they have been for years.)
Goals/Needs:	To graduate with above a 2.0 GPA and to party with his friends as much as possible before college ends. His goal is to graduate from U of A with above a 2.5 GPA, but he is not too sure if he can maintain his current party boy lifestyle and also get above a 2.5 GPA

Frustrations/Fears/Challenges:	Fred gets frustrated by how slow the dishwasher and washing machine work since there are so many people using them it would be nice if they were a bit faster. A challenge he faces is trying to find a time the washing machine isn't being used, or trying to fit his dishes in the dishwasher since it is always so full because it is so slow and can only do a few loads a day.
Preferred Channels:	Text (and only text)
Motivations:	He is motivated by the promise of a good job and eventually to take over at his dad's custom machinery company if he can graduate. He has made a deal with his dad that if he went to college and graduated with above a 2.0 GPA, then he would get a high ranking job right off the bat and eventually take over the company when his dad decided to retire.
How can we help?	We can help by giving Fred an easy way to request either servicing on the washing machine and dishwasher or a way to request upgrades in some of the old appliance at the house. This will only get used by Fred if it is entirely through an application so he doesn't have to take time out of his day to call people and organize a time to show his property manager the problems and deal with the service man and payments. It needs to be as simple as snapping a photo of the appliance maintenance or replacements are wanted, shooting the request off to the property manager with a few sentences about the problem and then being completely done (minus agreeing to a time with the service company) as the rest of the work will be done by the property manager.
Tech Acumen	Young enough to be a native tech user. Been using computers and phones his whole life.
Created By:	Adam Berard

Name:	Property manager George Gómez
Age:	45
Occupation:	Property manager of Valencia Apartments
Years as Property manager:	10 years
Hometown:	Boulder,CO
Character:	Strict, punctual, serious

Bio:	George decided to move to San Luis Obispo, California with his wife when he was 25 years old with no kids. He wanted a job that wouldn't take too much effort from him to manage while also making a decent amount of money. He was offered the position as property manager for his organized and strict personality that the owners appreciated. He has been doing it for about a decade now and now has a set of twins that will be graduating high school and entering college soon.
Goals/Needs:	George wants to minimize his workload as much as possible. He wants to focus more time towards his family as his kids are leaving for college soon. He will also need to maximize his revenue per time worked in order to pay for their college tuition and fees.
Frustrations/Fears/Challenges:	At the beginning of the school year, each year, he is overwhelmed with paperwork and maintenance requests on the apartments. He also has to manage with maintenance requests on his own. His twin daughters used to help manage some office business but, with them gone, he will need more help soon and he does not want to hire someone new.
Preferred Channels:	Prefers call or email. Text is inappropriate between him and his tenants
Motivations:	He is motivated to send his daughters to good schools while also keeping him and his wife financially stable for the upcoming year.
How can we help?	We can help by taking away the paperwork from his workload. We can also streamline tenant and property manager communication with community announcements, maintenance requests automation, and/or reports or issues.
Tech Acumen	Basic Technology Knowledge. Relies on his daughters for more advanced features.
Created By:	Cesar Chacon

Name:	Tenant Dwayne Martin
Age:	28
Occupation:	Assistant to the Regional Manager at Best Buy
Years Renting:	6 years
Hometown:	Newark,NJ

Character:	Diligent, focused, career-focused
Bio:	Dwayne was raised as a middle child of 7 children on a small farm house but always wanted to venture out and live in a more urban setting. With an interest in technology but unable to go to college, he started working at a Best Buy rising through the ranks hoping to one day be a manager. For many years after moving out of his parents house, Dwayne has always been plagued trying to find and maintain good housing in the city on a budget. He is now looking for anyway he can spend more time on his career and less on distractions.
Goals/Needs:	He wants to spend less time having to ask his forgetful property manager to perform repairs on his apartment and sometimes having to do the repairs himself. He is also annoyed by the fact he needs to take time off of work in order to show the repairmen what is broken in his apartment himself.
Frustrations/Fears/Challenges:	He is worried that the time he has to take off work in order to maintain his apartment is negatively affecting his opportunity to get a promotion. He is also frustrated having to call his property manager during work to remind her to find plumbers/electricians when necessary since otherwise he forgets and doesn't make service requests by the time he gets home from work.
Preferred Channels:	Cell phone or email
Motivations:	He is motivated above all to receive a promotion and eliminating any barriers getting in the way of this.
How can we help?	We can offer Dwayne a way to request repairs for his apartment that will ensure that the property manager won't forget what needs to be requested. We can also offer Dwayne a way for him to be able to go to work when a repair is being done by allowing him to show where in his apartment the repair needs to be done.
Tech Acumen	Adept at technology due to work at Best Buy.
Created By:	Tommy Bergmann

2.4 User Classes and Characteristics

User Class	**Description**

Property Manager	The property manager will be the user with the most permissions and actions available. They will be in charge of most of the transaction and may even initialize it on occasion instead of the tenant. They will accept or deny service requests
Tenant	The tenant is able to request repairs/replacements and help schedule what time the service company will come. They are also able to communicate with the property manager through the chat communication if the PM allows them too.

2.5 Operating Environment

The application will operate with React Native. Our frontend application will be using the Roopairs' Service Provider Network API as the backend and will also be integrated into web-based housing management software, PolyRents, in the future. Furthermore, we are using Expo CLI as our main build tool and project server to test and run our app. We are using Heroku for hosting our websites and PostGreSQL as our database to hold our model logic.

2.6 System Constraints

HomePairs must be built using the Roopairs API. It must also be able to integrate well with PolyRents' website which is built on React. Naturally, this means our application will be restrained to using React Native. We must make the application simple and easy to understand by request of the product owner and by general standard of UI/UX.

2.7 Assumptions and Dependencies

- We are dependent on complying with our software being open-sourced under the MIT license so that any property management software platform can use our open-source components to bring the power of Roopairs to their platform.
- We are dependent on complying with the Roopairs API connecting service providers with customers using property management software.
- We assume that financial transactions will either be addressed between the service provider and property manager or via the Roopairs API.
- We are dependent on allowing our application to be easily integrated PolyRents. To this end, we assume that they use React. We will develop our applications in React Native, which we assume to be compatible / easily integrated with React applications.

- We are dependent on our application being hosted on Github.
- We assume that our application needs to be able to function standalone despite being intended for integration with other property management software platforms. Functionality we expect to be replaced, such as account registration, will be fairly bare bones and self-explanatory, and not elaborated on in our use cases.

3 Use Cases

3.1 Use Case 1: Tenant request service on an Appliance

Use Case ID:	1
Use Case Name:	Tenant sends a Job Request to Property Manager
Created By:	Cesar Chacon
Last Updated By:	Chacon
Date Created:	October 6, 2019
Date Last Updated:	December 4, 2019
Actors:	Property Manager, Service Provider
Description:	A tenant's refrigerator has not been maintaining standard refrigeration temperatures. This has resulted in numerous food to spoil quicker than expected. The tenant goes onto the HomePairs app and requests for the appliance to be fixed to their Property Manager.
Preconditions:	1. Tenant has a HomePairs account connected to their Property Manager's account. 2. Property manager has logged a list of appliances maintained on Roopairs. 3. The refrigerator is unable to maintain standard refrigeration temperatures.
Postconditions:	1. The refrigerator is now functioning properly.
Normal Flow:	1.0 Tenant Requests Service for an appliance and it is approved

	1. Tenant logs onto HomePairs 2. Tenant searches through the list of appliances. 3. Tenant selects the refrigerator. 4. Tenant chooses to request service on this refrigerator 5. Tenant enters the details of what is wrong with the refrigerator. 6. Tenant sends the request to the Property Manager to be approved or denied 7. The request is approved and a service provider is scheduled to fix the appliance
Alternative Flows:	1.1 Tenant Requests Service for an appliance and it is denied (branch after step 6) 1. Property manager declines tenant's request for a service of choice. 2. Return to step 2.
Exceptions:	1.0.E.1 A Property manager or tenant cancelling the request and a service company requesting a cancellation fee.
Priority:	High
Frequency of Use:	Thousands of users per day.

3.2 Use Case 2: Communication Between Property Managers and Tenants.

Use Case ID:	2
Use Case Name:	Communication Between Property Managers and Tenants.
Created By:	Eeron Grant
Last Updated By:	Eeron Grant
Date Created:	October 6, 2019
Date Last Updated:	October 20, 2019
Actors:	Property Manager, Tenant
Description:	This use case is to give the parties involved with requesting, approving, and receiving services a means of communication. This will properly keep all users informed and effectively reduce communication barriers.
Preconditions:	1. Tenants and Primary Property Manager already have HomePairs account.
Postconditions:	1. The chat box data is still preserved.

Normal Flow:	2.0 Property manager Sends Message to Tenants 1. Property manager chooses to create a chatroom. 2. Property manager is prompted to allow users to post messages in chat-room 3. Property manager chooses to disable users from submitting messages 4. Property manager is provided a list of managed properties. 5. Property manager selects a property. 6. Property manager chooses individuals or chooses to add all tenants in that property. 7. Property manager is navigated back to the list of different properties. 8. Property manager repeats step 4. 9. Property manager chooses a name for the chat. 10. Property manager sends a message. 11. All tenants in the chat room receive the message.
Alternative Flows:	2.1 Property manager removes a selected tenant from being an addition to the chat (branch after step 5) 1. Property manager repeats step 4 and 5 2. Property manager selects individual to remove 3. Property manager is navigated back to step 4 2.2 Property manager Allows Users to Submit Messages (branch after step 2) 1. Property manager allows users to submit messages 2. Property manager continues to step 4
Exceptions:	2.1.E Property manager Cancels Chat Room 1. Property manager clicks button to cancel chat room 2. Property manager is navigated back to home page
Includes:	None
Priority:	High
Frequency of Use:	Approximately Thousands of Users
Business Rules:	TBD
Assumptions:	Assumes that all Property managers and tenants have access to some network.
Notes and Issues:	TBD

3.3 Use Case 3: Property manager Approving and Sending Service Requests to Roopairs Network

Use Case ID:	3
Use Case Name:	Property manager Approving and Sending Service Requests to Roopairs Network.
Created By:	César Chacon
Last Updated By:	Eeron Grant
Date Created:	October 7, 2019
Date Last Updated:	October 20, 2019
Actors:	Property Manager, Tenants
Description:	The property manager has received a service request to fix a refrigerator from one of his tenants. The property manager views the request and sends the requests to the Roopairs service providers network and receives back 5 recommendations.
Preconditions:	1. The property manager and tenant have not yet communicated about the problem with the refrigerator.
Postconditions:	1. The property manager has sent the job requests to Roopairs and has chosen a service provider to handle the job.
Normal Flow:	3.0 Service is Requested and Successfully Completed 1. Tenant sends request for a general service or specific appliance. 2. Property Manager receives a notification about the service request and sends it to the Roopairs network. 3. Roopairs responds with 5 vetted service providers 4. Property Manager looks over each of the service providers profiles and/or looks through favorite service providers they have worked with before. 5. Property Manager chooses a service provider.
Alternative Flows:	3.1 Property Manager Cancels Tenants Request (branch after step 2)
Exceptions:	None

Special Requirements:	1. The tenant cancels the request for a service after the request has been made. 2. The property manager denies a service request. 3. The service provider may cancel/pause a job.
Includes:	None
Priority:	Medium
Frequency of Use:	Per user, once or twice a year.
Business Rules:	TBD
Assumptions:	Assume that 30 tenants and property managers already have chat-rooms amongst each other.
Notes and Issues:	TBD

3.4 Use Case 4: Viewing the Status of a Service Job.

Use Case ID:	4
Use Case Name:	Viewing the Status of a Service Job.
Created By:	Tommy Bergmann
Last Updated By:	Eeron Grant
Date Created:	October 7, 2019
Date Last Updated:	October 20, 2019
Actors:	Property Manager, Tenant
Description:	Being able to see the status of a service has many applications. From a tenant eager to see the status of their washer to a property manager wanting to know if the servicemen is about to arrive to the destination of the request.
Preconditions:	1. The tenant does not know the progress of their service request.
Postconditions:	1. The tenant knows exactly which step of the process the service request is on.
Normal Flow:	4.0 User Successfully Views the Status of their Request 1. The tenant logs into their HomePairs account. 2. The tenant navigates to their active requests. 3. The tenant selects the request they wish to view. 4. From this page, the tenant views precise status of the request. 5. The tenant takes action, if necessary.
Alternative Flows:	4.1 A Completed Service is Viewed (branch after step 3)

	1. From step 3 of 4.0, the service request is marked as completed.
	4.2 A Service is Canceled (branch after step 3) 1. From step 3 of 4.0, the service request is marked as canceled.
Exceptions:	4.0.E.1 Service Denies Request (at step 4) 1. If the job is paused or cancelled for any reason, the property manager and tenant should be notified by their preferred channel of communication of this and the status of the job should be updated to "Cancelled" or "Paused".
Special Requirements:	1. Request is canceled. 2. HomePairs displays which party canceled
Includes:	None
Priority:	Medium
Frequency of Use:	At least once per a service request
Business Rules:	TBD
Assumptions:	TBD
Notes and Issues:	TBD

3.5 Use Case 5: Property Manager and Tenant Repair an Appliance

Use Case ID:	5
Use Case Name:	Property Manager and Tenant Repair an Appliance
Created By:	Sarah MacDougall
Last Updated By:	Sarah MacDougall
Date Created:	October 8, 2019
Date Last Updated:	October 20, 2019
Actors:	Property Manager, Service Provider, Tenants
Description:	The property manager has designated his property manager to manage service requests on his behalf while he is out of the country. The property manager received a service request to fix a toilet from one of his tenants. The property manager views the request and sends the requests to the Roopairs service providers network and hires their favorite plumber.

Preconditions:	1. The property manager has granted the property manager repair privileges. 2. The property manager won't be able to check their phone for a while. 3. The tenants and property manager have not yet communicated about the problem with the toilet. 4. The property manager has a favorite plumber he prefers to hire.
Postconditions:	1. The property manager and service provider have processed their job request through HomePairs. 2. The property manager can view the updated status of the job / job request and see their conversation and receipts.
Normal Flow:	5.0 Service is Requested and Successfully Completed 1. Tenant sends request for service. 2. Property manager receives a notification about the job request. 3. Property manager can view all jobs that were performed previously on a certain appliance. 4. The property manager selects his favorite plumber and sends them a job request through the repairs network. 5. The service provider accepts the job request, and the property manager can see the job request has updated to a job that has been started. 6. The property manager is notified when the job is completed and pays the service provider through HomePairs. 7. The property manager finally checks their phone again and has a notification about a new completed job, and is given a description of the job request, the service provider hired, and a receipt from the service company.
Alternative Flows:	5.1 Property Manager Denies Tenants' Request (branch after step 2)

	1. The property manager starts a chat with the tenants, saying that he used a plunger and that the toilet appeared to be working as good as new. 2. The property manager finally checks their phone again and has a notification about a denied job request, is given a description of the job request, and is given access to the related chat.
	5.2 Service Provider Fails to Complete the Job Request (branch after step 4) 1. The property manager is notified that the job's status has been stopped and sees that the service provider has started a chat, saying that he is unable to repair the toilet and that they will need a new one. 2. The property manager responds, saying that he will wait for the property manager to decide if he is fine with replacing the toilet instead of repairing it. 3. The property manager responds, accepting the new terms of the Job. (resume from step 5)
Exceptions:	None
Special Requirements:	1. The tenant cancels the request for a service after the request has been made. 2. The property manager denies a service request. 3. The service provider may cancel/pause a job.
Includes:	None
Priority:	Low
Frequency of Use:	Once every couple years.
Business Rules:	TBD
Assumptions:	Assume that the property manager has already granted the property manager property manager privileges and that the property manager and tenants already have chat-rooms amongst each other.
Notes and Issues:	TBD

3.6 Use Case 6: Users Notified of the Status of a Job Through Completion

Use Case ID:	6

Use Case Name:	Users Notified of the Status of a Job Through Completion
Created By:	Eeron Grant
Last Updated By:	Eeron Grant
Date Created:	October 19, 2019
Date Last Updated:	October 20, 2019
Actors:	Property Manager,Service Provider, Tenants
Description:	Jobs will undergo different stages of completion. The primary users need to have have some level of awareness of the status of these jobs. When using the HomePairs app, users will be notified when the status of a job has been changed.
Preconditions:	1. Service provider has a profile to be able to change the status of a job. 2. A job request has been sent from the tenant to the property manager.
Postconditions:	1. All users receive relevant alerts indicating the status of a job. 2. The job is marked as complete.
Normal Flow:	6.0 Service is Accepted at All Stages and Successfully Completed 1. Property manager receives a notification about the job request. 2. The property manager selects service for the request through the Roopairs network. 3. The service provider accepts the job request. 4. The status of the job is changed from Pending to Progress. 5. Property manager and tenants of the property are notified of the accepted request. 6. The service provider provides the service. 7. The service provider marks the job as Completed. 8. The property manager and tenants are notified of the Jobs change of status. 9. The property manager and tenants are prompted to review the Job and the service provider.
Alternative Flows:	6.1 Property Manager Denies Tenants' Request (branch after step 1)

	1. The property manager denies tenants' request for service. 2. Tenants are notified of request status. 6.2 Service Provider Denies Request (branch after step 2) 1. The service provider denies the job request. 2. The property manager and tenants are notified of canceled request. 3. Repeat step 2
Exceptions:	6.0.E.1 Service is Canceled At Any Point 1. The Service Provider cancels the job after accepting a job request. 2. The status of the job is marked as canceled. 3. The property manager and tenants are notified of the change of status.
Special Requirements:	1. The tenant cancels the request for a service after the request has been made. 2. The property manager denies a service request. 3. The service provider may cancel/pause a job.
Includes:	None
Priority:	Low
Frequency of Use:	Once every couple years.
Business Rules:	TBD
Assumptions:	Assume that the property manager has already granted the property manager property manager privileges.
Notes and Issues:	TBD

3.7 Use Case 7: Viewing All Jobs Performed on an Appliance

Use Case ID:	7
Use Case Name:	Viewing All Jobs Performed on an Appliance
Created By:	Eeron Grant
Last Updated By:	Eeron Grant
Date Created:	October 19, 2019
Date Last Updated:	October 20, 2019
Actors:	Property Manager, Tenants

Description:	At times, users may want to view the history of the different services provided on an appliance. They may require information about the previous Service Providers, the date of the last job, or even if the job had been completed successfully.
Preconditions:	1. There are appliances defined in the HomePairs app for the properties associated with the user.
Postconditions:	None
Normal Flow:	7.0 A Property Manager Views the History of an Appliance 1. The property manager navigates to the properties page. 2. The property manager is displayed a list of properties. 3. The property manager chooses a property. 4. The property manager is navigated to a list of Tenants and Appliances associated with that property. 5. The property manager selects an Appliance. 6. Information about the Appliance is displayed. 7. The service provider scrolls down the page to view the history of services applied to the Appliance.
Alternative Flows:	7.1 A Tenant Views the History of an Appliance (branch after step 1 in place of property manager) 1. The tenant navigates to the Appliances page. 2. The tenant is displayed a list of appliances associated with their home. 3. Tenants continue the flow from step 6 in place of the property manager.
Exceptions:	None

Special Requirements:	1. The user must be able to navigate backwards from the page. 2. The user must be able to use the navigation bar to change pages. 3. If an Appliance has not had any services conducted for it, it will not display a history. 4. The flow should also apply to Building Systems and Building Spaces in the same order but in the place of Appliances.
Includes:	None
Priority:	Medium
Frequency of Use:	Every few weeks (Tenants) to daily (Property Manager).
Business Rules:	TBD
Assumptions:	Appliances of a particular property are view-able by the tenants living in that property.
Notes and Issues:	TBD

3.8 Use Case 8: Viewing Past and Current Jobs

Use Case ID:	8
Use Case Name:	Viewing Past and Current Jobs
Created By:	Eeron Grant
Last Updated By:	Eeron Grant
Date Created:	October 19, 2019
Date Last Updated:	October 20, 2019
Actors:	Property Manager, Tenants
Description:	This is a simple case in which users of the HomePairs app may desire to view past/current jobs.
Preconditions:	1. The users have properties associated with their HomePairs account.
Postconditions:	None
Normal Flow:	8.0 A User Views the Details of a Past Job

	1. The User navigates to the Services page. 2. The User is displayed a list of services organized by completion date, request date, status, and (if applicable) property. 3. The User scrolls down to a past job. 4. The User selects the past job. 5. Details of the past job are displayed to the User.
Alternative Flows:	8.1 The User Filters the list of Jobs (branch after step 2) 1. The user inputs parameters into a search bar. These parameters can be found in Special Requirements. 2. A subset of Jobs that exist in the domain of the search are displayed. 3. The User continues to step 3.
Exceptions:	None
Special Requirements:	1. Parameters for a search are from Appliance, Property, Services, Start Date, Completion Date, and Status. 2. The user must be able to use the navigation bar to change pages. 3. If no job has been requested by a particular account, the display will output that no Job has been completed. 4. If all jobs do not meet the search parameters, then the display will output that no job has been found.
Includes:	None
Priority:	Medium
Frequency of Use:	Every few weeks (Tenants) to daily (Property Manager).
Business Rules:	TBD
Assumptions:	None
Notes and Issues:	It may be difficult to properly implement a search function with a search bar. This is more of a stretch goal to hopefully implement a simple approach to filtering jobs.

4 System Features

4.1 System Feature 1

Roopairs Service Request and Approval - Specific Service Provider

4.1.1 Description and Priority

A high priority feature that must be included. This feature allows users to request services that will get approved by a higher authority (Property Manager). Once approved, the system will send the request to the service provider/providers that the property manager has previously designated as preferred service providers. The service provider will then have the opportunity to accept the job. Once service has been provided, the property manager and service provider will be prompted to complete the status of the job.

4.1.2 Stimulus/Response Sequences

Upon receipt of a request for service from a tenant, if the property manager thinks the repair is valid then the system shall respond by sending the request to the Roopairs network.

4.1.3 Functional Requirements

4.1.3.1 General

1. GEN-1: The system shall allow users to be able to view real time status of Jobs and Job Requests.
2. GEN-2: The system shall allow users to view current and past service jobs.
3. GEN-3: The system shall notify users when a Job or Job Request changes status.

4.1.3.2 Property Manager

1. PM-1: The system shall allow property managers to view their current residents and the addresses of their properties.
2. PM-2: The system shall allow property managers to approve and send service requests to Roopairs network.
3. PM-3: The system shall allow property managers to triage (rank urgency) service requests from tenants.
4. PM-4: The system shall allow property managers to search for and hire service companies for approved repair requests.

5. PM-5: The system shall allow property managers to cancel a paused job before it has been officially started.

6. PM-6: The system shall allow property manager to make payments powered by Roopairs through the application.

7. PM-7: The system shall allow property managers to view reports and analysis on Jobs.

8. PM-8: The system shall allow property managers to either send their request to their preferred service companies, or have the Roopairs network give them recommendations (which will have a much faster response time).

4.1.3.3 Tenant

1. TEN-1: The system shall allow tenants to request appliance repairs, appliance replacements, and appliance upgrades from their property manager.

2. TEN-2: The system shall allow tenants to track jobs as they are in progress and receive updates from the property manager.

3. TEN-3: The system shall allow tenants to select an existing appliance in their JobRequest or be able to create / suggest / specify an appliance that has not been added yet.

4.2 System Feature 2

Roopairs Service Request and Approval - With Service Provider Options

4.2.1 Description and Priority

A high priority feature that must be included. This feature allows users to request services that will get approved by a higher authority (property manager). Once approved, the system will send the request to the Roopairs API and get back a list of potential service providers to choose from. The property manager will then select one of the service providers to complete the job. Once service has been provided, the property manager and service provider will be prompted to complete the status of the job.

4.2.2 Stimulus/Response Sequences

Upon receipt of a request for service from a tenant, if the property manager thinks the repair is valid then the system shall respond with a sending the request to the Roopairs network.

4.2.3 Functional Requirements

4.2.3.1 Property Manager

1. PM-1: The system shall allow property managers to be presented with the recommended service providers that the Roopairs API provides as well as their favorite service providers.
2. PM-2: The system shall allow property managers to easily look through the profiles of the recommended service providers.
3. PM-3: The system shall allow property managers to access information easily about each recommended service provider so they can easily decide on the best service provider.
4. PM-4: The system shall allow property managers to easily add a service provider to their preferred list of service providers after a job has been completed.
5. PM-5: The system shall allow property managers to be presented with an easy way to compare the recommended service providers to help them make their decision.

4.3 System Feature 3

Property manager, Tenant, and Service Provider Communication via HomePairs Messaging

4.3.1 Description and Priority

A medium priority feature that has much value and should be included. This feature allows property managers, tenants, and service representatives to have a means of communication via the HomePairs platform during the appropriate contexts. This will keep communications about housing and services within the application.

4.3.2 Functional Requirements

4.3.2.1 General

1. GEN-1: The system shall allow users to create and access a chat group for every job and the chats will be referenced by their job title.

4.3.2.2 Property Manager

1. PM-1: The system shall allow property managers to view their current residents.
2. PM-2: The system shall allow property managers to open conversations with their tenants in the app.

3. PM-3: The system shall allow property managers to open conversations with hired service companies.
4. PM-4: The system shall allow property managers to view any conversations and receipts from Jobs managed by their Property Managers.
5. PM-5: The system shall allow property managers to easily send pre-filled out messages to their tenants. For example, after deciding on a price a pre-filled message could come up that says "Update: Larry's Lawnmower Fixers have been selected as the service company that will fix the lawn mower you requested. What days work best for you out of: [list of available days]."

4.3.2.3 Tenants

1. TEN-1: The system shall allow tenants to communicate with the service company to set up a time for them to come.

4.4 System Feature 4

Build and Display Service Provider Profiles

4.4.1 Description and Priority

A medium-high priority feature that has much value and should be included. This feature gives service providers a profile with data from the API that can be used by potential clients to see some information about them, as well as reviews and possibly a picture and their prices.

4.4.2 Functional Requirements

4.4.2.1 General

1. GEN-1: The system shall create a profile for each service provider.
2. GEN-2: The system shall allow users to view service providers profiles before they decide on who to hire.
3. GEN-3: The system shall allow users to review service providers when the job has been completed.
4. GEN-4: The system shall include service providers' rating, previous jobs, reviews, general information, and contact information in their profile.

4.5 System Feature 5

Add and Manage Appliances

4.5.1 Description and Priority

A high priority feature that has much value and should be included. This feature allows both property managers and tenants to log appliances present at the property so that when service is needed it is easy to select the correct appliance.

4.5.2 Functional Requirements

4.5.2.1 Property Manager

1. PM-1: The system shall allow property managers to log appliances (oven, fridge, toilet, gutters, garage door), including photos, their location, a description.
2. PM-2: The system shall allow property managers to be able to choose to save a new appliance submitted by a tenant.
3. PM-3: The system shall allow property managers to see all the appliances from each of their properties and sort by different properties such as most repairs done on, time since last repair, age of appliance, etc.

4.5.2.2 Tenant

1. TEN-1: The system shall allow tenants to send Job Requests for appliances to their property manager, including photos and descriptions.
2. TEN-2: The system shall allow tenants to select an existing appliance in their Job Request or be able to create / suggest / specify their own appliance.
3. TEN-3: The system shall allow tenants to create an appliance and then have it sent to their property manager for approval.

5 External Interface Requirements

5.1 User Interfaces

The design of the UI will be focused on creating an interface that should be easy to use, even for people that do not use technology often. This includes considerations like buttons being large and labeled and keeping each screen simple (non-cluttered).

5.2 Hardware Interfaces

Since HomePairs will be a web-app, any device that can connect to the website should be able to use the application. Consequently, we will have to make UI designs for desktop vs. mobile version of our site. One important consideration while making the two versions is that the UI between desktop and mobile should be very similar to avoid any user confusion.

5.3 Software Interfaces

The application will be developed using React Native, and integrated with the Roopairs API. In order to store user data (credentials, message history), we will be using PostGreSQL as our database platform.

5.4 Communications Interfaces

There are several different types of communication in the HomePairs application. This includes the communication between property managers and service providers that is handled by the Roopairs API. Also, the messaging system handles communication between a combination of property managers, property managers, tenants, and service providers. The last type of communication is tenants requesting a service from their property manager.

6 Nonfunctional Requirements

6.1 Performance Requirements

1. Our application will show 5 recommendations received from the Roopairs API immediately (within 5 seconds to load) after the request is sent.
2. Our application should reflect status updates to users within 5 minutes of Roopairs creating such updates.
3. Our application will send and receive messages immediately between two or more users.
4. Our application should be able to hold chat rooms that can hold can hold a maximum of 200 people.

6.2 Security Requirements

1. Our application must facilitate the payment through Roopairs and Stripe and ensure that we do not store any sensitive data.
2. Our application must store sensitive data, such as chat messages and passwords, securely. This sensitive data must be stored in the database using encryption rather than plain text.
3. Our application must query databases securely.
4. Our application must limit visibility of passwords fields.
5. Our application must support two factor authentication.

A Glossary/List of Terms

Appliance: A device or piece of equipment designed to perform a specific task in a building unit.
Building System: A foundation installed into a building unit that provides a specific need. Examples include electrical systems and plumbing systems.
Building Spaces: Spaces in a building reserved for a common purpose. Kitchens, bedrooms, and washrooms are all examples of Building Spaces.
Chat: The communication/messages between two users.
Chat Rooms: A defined list of users that are capable of sending messages amongst each other.
DashBoard: The focus of the HomePairs home page. It show cases current most recent statuses of jobs associated with the user. It also is the main page for navigation throughout the HomePairs application.
Job: A task that has been received, accepted, and currently be addressed by a Service Provider.
Job Request: A proposal for receiving service sent from the tenant to Property Manager, and from the Property Manager to service. These can be approved or denied from the receiving party.
Navigation (NAV) Bar: A component that sits at the top or the bottom of a user interface that contains various buttons that navigate a user to different components of the application.
PolyRents: An application whose primary use cases are to simplify the leasing process nearby Cal Poly by digitizing the housing application process.
Progressive Web App (PWA): A mobile friendly website application that has the capability of functioning offline.
Property Manager: Someone with authority over at least one property in need of service. The user who will determine to accept requests from tenants for services. (Stretch Goal - They can also initiate chat channels.)
Real-time Status: The actual time during which something takes place. In the HomePairs app status will show Pending, Progress, and Completed.
Residential Sector: The potential market for this application. This market demographic mostly consists of homeowners and renters.
Roopairs API: The application programming interface that will conduct a majority of the business logic as constrained by the our customer -Roopairs
Service Provider: A company/individual that will give service to users. They will be listed from the Roopairs API.
Tenant: A user of the HomePairs application that is leasing a property from a property manager.
Web Application: An application that is built on a device which has a UI built specifically for web browsers. It is divided into front-end (browsers) and back-end (servers) to allow remote usability on the web.