Project Name: Roopairs
Software Requirements Specification
version 2.0

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

October 23, 2019

Contents

Credits

Name	Date	Role	Version
Luke Reckard	October 10, 2019	Document Owner, Co-Author of Software Requirements Specification	1.0
Karla Sunjara	October 10, 2019	Co-Author of Software Requirements Specification	1.0
Jessica Chang	October 10, 2019	Co-Author of Software Requirements Specification	1.0
Yusuf Bahadur	October 10, 2019	Co-Author of Software Requirements Specification	1.0
Logan Lawson	October 10, 2019	Co-Author of Software Requirements Specification	1.0

Revision History

Name	Date	Reason for Changes	Version
Karla Sunjara	October 22, 2019	Updating use cases and functional requirements	2.0
Yusuf Bahadur	October 22, 2019	Updating use cases and interface requirements	2.0
Logan Lawson	October 22, 2019	Updating use cases and functional requirements	2.0
Luke Reckard	October 22, 2019	Updating use cases and functional requirements	2.0
Jessica Chang	October 22, 2019	Updating non-functional requirements and overall description	2.0

1 Introduction

1.1 Purpose

This document presents the requirements and limitations for the Repairs version 1.0 that will be available on the point of sale system called Clover. It will serve as a guidance to the specified parties throughout the development process. It will cover major features, constraints of the final product, and any non functional requirements that should be met.

1.2 Document Conventions

This document shall adhere to the following formatting conventions:

- References to other documents shall be underlined and italicized.
- Important terms are defined in the glossary at the end of the document.

1.3 Intended Audience and Reading Suggestions

1.3.1 Developers

The development team will reference this document throughout the design and implementation of this application. Their main points of interests are the User Cases, System Features, and Nonfunctional Requirements to guide decisions throughout the development process.
Suggested Reading Order

1. Overall Description
2. Use Cases
3. System Features
4. External Interface Requirements
5. Other Nonfunctional Requirements

1.3.2 Project Owner - Alexander Kavanaugh, David Bartolomucci, Ray Bartolomucci

Alex, the team's connection with Roopairs and the CTO of Roopairs, will use this document to ensure that the development team understands the key features of the end product. David will help with the understanding of the features and business use cases. Ray will help our team with understanding Clover interactions.
Suggested Reading Order

1. Overall Description
2. Use Cases
3. System Features
4. External Interface Requirements
5. Other Nonfunctional Requirements

1.3.3 Supervisor - Dr. David Janzen

Dr. Janzen, the team's professor in CSC 402, will review this document to obtain a better understanding of the team's goals for the final application. Furthermore, he will reference this document throughout the development process to remain up to date with any major changes in the features or scope.
Suggested Reading Order

1. Overall Description
2. System Features
3. External Interface Requirements
4. Other Nonfunctional Requirements

1.4 Project Scope

The main goal for this project is to connect restaurant owners with service professionals and allow restaurant owners to better manage their own equipment.

For further information, reference the teams *Vision and Scope* document.

https://www.overleaf.com/6362151365nrkrwqzkmtdq

2 Overall Description

2.1 Product Perspective

The restaurant industry is plagued with the problem of finding an equipment repair professional on short notice when essential restaurant equipment breaks down during operating hours. Developing an application that can integrate with a restaurant's existing POS (Point of Sale) system's platform, specifically Clover, will allow employees to access a large database of available service professionals. The ease of this app will make the searching and requesting for immediate service smoother and more efficient for restaurants.

2.2 Product Features

FE-1	User can submit service requests when the restaurant's equipment breaks down.
FE-2	User can log and keep track of their restaurant's equipment.
FE-3	User can view previous jobs for each piece of restaurant equipment.
FE-4	User can view the invoices of all past service jobs.
FE-5	User can receive analysis from their past services for each piece of equipment in order to help the user make more informed decisions in the future.
FE-6	The application will have different levels of permissions for staging service requests depending on a user's position at the restaurant.

2.3 User Classes and Characteristics

User	**Description**
Developer	Designs and builds the application. The developer will also test the application and ensure that it integrates well with the Clover system.
Customer (Roopairs)	Communicates to developers what is needed in the application. Discusses and prioritizes the most important features.

2.4 Operating Environment

Our operating environment will be the Clover POS system. The application is meant to be downloaded from the Clover App Marketplace and used on the Clover device, which is a touch screen tablet.

2.5 Design and Implementation Constraints

Much of the constraint for this project will be the developers skill level and knowledge. No one has experience with React Native and a majority of us do not have experience with JavaScript.

2.6 User Documentation

Once implementation begins, developers will be keeping track of all progress using JIRA. Documents that will be produced during the design of the application include a SRS, a software architecture document, and UML diagrams. The text documents will be formatted in LaTex. Using the produced documentation as guidelines, we will develop prototypes.

2.7 Assumptions and Dependencies

It is assumed that the Roopairs API will be made available for the developers during prototype development. It is assumed that the user will know how to use the Clover POS system.

2.8 Functional Requirements

FR-1a	System shall allow the user to create a service request
FR-1b	System shall allow the user to cancel a service request.
FR-1c	System shall allow the user to view the details of a service request.
FR-1d	System shall allow the user to select a flexible or urgent repair time frame.
FR-2	System shall allow the user to select a service provider for a service request.
FR-3a	System shall allow the user to store details about the restaurant equipment.
FR-3b	System shall allow the user to view the details about restaurant equipment.
FR-3c	System shall allow the user to filter restaurant equipment based on category.
FR-3d	System shall allow the user to see past service requests for a piece of equipment.
FR-4	System shall allow the user to view all past service requests.
FR-5	System shall allow the user to register a Roopairs account through the application.
FR-6	System shall allow the user to log in to the application.
FR-7	System shall allow the user to change the restaurant location.
FR-8	System shall allow the user to create a list of preferred service providers.
FR-9a	System shall allow the user to invite service providers to the Roopairs platform.
FR-9b	System shall allow the user to edit the list of preferred service providers.

3 Use Cases

3.1 Use Case 1: Create a Service Request

Use Case ID:	1
Use Case Name:	Create a Service Request

Created By:	Karla Sunjara
Last Updated By:	Karla Sunjara
Date Created:	October 3, 2019
Date Last Updated:	October 22, 2019
Actors:	Users
Description:	The user clicks on the add service request option for a repair category and inputs all of the necessary data.
Preconditions:	1. User is logged into the application. 2. User has the correct permissions to send the request.
Postconditions:	1. A confirmation email is sent to the restaurant.
Normal Flow:	1.0 Create a Service Request 1. User clicks on the add service request for an appliance option. 2. User clicks on the piece of equipment that needs repair. 3. System displays input fields the user must fill in such as preferred service provider and flexible or urgent repair time frame. 4. User inputs data and clicks the send service request button. 5. System sends the request to the Roopairs service. 6. System displays a success screen. 7. User clicks on "OK" button and returns to the home page.
	1.1 User wants to edit the service request before approval (after step 6) 1. User clicks on the service request details button on the home page. 2. User clicks edit request and changes the necessary data fields. 3. User clicks on update request. 4. Return to step 5. 1.2 User selects an urgent repair time frame (after step 3)

	1. System displays to the user a message stating that extra charges may come with the urgent repair time frame. 2. Return to step 3. 1.3 User wants to cancel the service request before approval (after step 6) 1. User clicks on the service request details button on the home page. 2. User clicks cancel request. 3. System displays a confirmation that the user would like to cancel the request. 4. User clicks on the "OK" button. 5. System removes request from the schedule. 6. System sends cancelled request to the Roopairs service. 7. Return to step 6. 1.4 User inputs a preferred service provider (after step 3) 1. User selects a preferred service provider. 2. Return to step 4.
Exceptions:	1.0.E.1 Required information is not provided (at step 3) 1. System informs the user that the necessary fields need to be filled out. 2. User inputs the required information and clicks the send service request button. 3. Return to step 4.
Includes:	None
Priority:	High
Frequency of Use:	High. Very frequent because this is the main functionality of the Repair application
Special Requirements:	1. User should be able to cancel the service request at any time prior to a company sending a quote. 2. User should be able to view all previous repairs on each equipment piece and repeat one of those service requests as the new request (Priority = high) 3. User should only be able to select one preferred provider.

Assumptions:	Assume that a piece of equipment will break down and need servicing
Notes and Issues:	1. The default date for the request is the current date.

3.2 Use Case 2: Accept a Pending Service Request

Use Case ID:	2
Use Case Name:	Accept a Pending Service Request
Created By:	Luke Reckard
Last Updated By:	Luke Reckard
Date Created:	October 22, 2019
Date Last Updated:	October 22, 2019
Actors:	Users
Description:	The user accepts a repair request from a service providers pop-up.
Preconditions:	1. User is logged into the application. 2. The user has the correct permissions to accept the request.
Postconditions:	1. A confirmation email is sent to the restaurant. 2. A confirmation email is sent to the service provider.
Normal Flow:	1.0 Accept a Pending Service Request 1. Service provider accepts a service request. 2. Roopairs API sends a service acceptance request to the system. 3. System displays a service request pop-up that contains a list of service providers for a submitted request. 4. User chooses one of the available service providers. 5. System displays the company, quote, and estimated time. 6. User accepts the service approval request. 7. System stores the service request in the database and displays the request as scheduled. 8. System sends confirmation to the Roopairs service.
Alternative Flows:	1.1 User cancels service request (branch after step 7)

	1. User cancels the service request. 2. The system notifies the service provider of the cancellation. 3. The system stores the cancellation in the database. 1.2 User wants to edit the service request after approval (after step 5) 1. User clicks on the edit service request. 2. System updates the request for the service provider. 3. Service provider updates any necessary information. 4. Return to step 1. 1.3 User inputs a preferred service provider and the preferred service provider is not available (before step 1) 1. The preferred service provider is not available and declines the request. 2. Roopairs API sends a declined request to the system. 3. System notifies the user that their preferred service provider is not available.
Exceptions:	1.0.E.1 User clicks done without selecting a service provider (at step 3) 1. System informs the user that a service provider must be selected 2. Return to step 2.
Includes:	None
Priority:	High
Frequency of Use:	High. After users create a service request, this is the next step in the pipeline that users will interact with.
Special Requirements:	1. If a user cancels a service request after a company sends a quote, the user will be charged a cancellation fee, unless they cancel 24 hours prior to the repair.
Assumptions:	Assume that a service request will receive quotes from service providers.

Notes and Issues:	1. It is unspecified about the communication between this application and the Roopairs application when there are no service providers.

3.3 Use Case 3: Input Equipment Information

Use Case ID:	3
Use Case Name:	Input Equipment Information
Created By:	Jessica Chang
Last Updated By:	Jessica Chang
Date Created:	October 7, 2019
Date Last Updated:	October 22, 2019
Actors:	Users
Description:	The user adds a new equipment to the application.
Preconditions:	1. User is logged into the application. 2. User is on the equipment page. 3. The user has the correct permissions to add appliances.
Postconditions:	1. User can view the new equipment's information through the application.
Normal Flow:	1.0 Add an equipment through the application 1. User clicks on add equipment option within the application 2. System displays texts fields like name of the equipment, serial number, make and model, and where it is located in the kitchen for user to fill out. 3. User inputs the data and clicks the save button. 4. System saves the equipment data in the database.
Alternative Flows:	1.1 User cancels adding an equipment (branch after step 3) 1. User clicks the cancel button. 2. The system returns to the equipment page. 1.2 User adds an equipment that's already been added (branch after step 4)

	1. The system notifies the user that the equipment has already been added. 2. The system returns to the add equipment page in case the user input the wrong information.
Exceptions:	1.0.E.1 Some information not provided (at step 4) 1. System informs user that the necessary fields need to be filled out. 2. User puts in required information. 3. Return to step 4.
Includes:	None
Priority:	High
Frequency of Use:	Not very frequent. Will most likely be used at the beginning, when user first starts the application or when the user purchases more equipment for their restaurant.
Special Requirements:	None
Assumptions:	Assume that the restaurant owners will want to input their equipment in case the equipment breaks down.
Notes and Issues:	None

3.4 Use Case 4: View Past Service Requests

Use Case ID:	4
Use Case Name:	View Past Service Requests
Created By:	Logan Lawson
Last Updated By:	Logan Lawson
Date Created:	October 8, 2019
Date Last Updated:	October 8, 2019
Actors:	Users
Description:	The user can view past service requests/work history for restaurant equipment.
Preconditions:	1. User is logged into the application. 2. User is on the restaurant equipment page.
Postconditions:	N/A
Normal Flow:	1.0 View past work history for all restaurant equipment through application.

	1. User clicks on the past work option of the restaurant equipment page. 2. System displays past service requests in a date ordered table structure. 3. User clicks on a past service request. 4. System displays full invoice for that service request.
Alternative Flows:	1.1 Viewing past work history for individual restaurant equipment through the application (at step 1). 1. User clicks on the piece of restaurant equipment they want to view. 2. System displays all details of the piece of restaurant equipment. 3. User clicks on the option to view past service requests. 4. System displays past service requests for that piece of equipment in a table ordered by descending date. 5. User clicks on a past service request. 6. System displays full invoice for that service request.
Exceptions:	1.0.E.1 No past service requests (at step 2). 1. System displays to user that there is no past service requests. 1.1.E.1 No past service requests (at step 4). 1. System displays to user that there is no past service requests.
Includes:	None
Priority:	High
Frequency of Use:	Medium. Will most likely be used when a piece of equipment breaks down and a manager is reviewing past service history of that equipment.
Special Requirements:	None
Assumptions:	Assume that the restaurant owners/managers will want to keep track of what equipment breaks and how often that equipment is out of service.
Notes and Issues:	None

3.5 Use Case 5: Register an Account

Use Case ID:	5
Use Case Name:	Register an account
Created By:	Yusuf Bahadur
Last Updated By:	Yusuf Bahadur
Date Created:	October 17, 2019
Date Last Updated:	October 17, 2019
Actors:	Users
Description:	The user creates a Roopairs account to use the application.
Preconditions:	1. Application is connected to a network.
Postconditions:	1. System displays a welcome animation to alert the user of successful registration. 2. System displays the main application page.
Normal Flow:	1.0 Enter User Information for Registration. 1. User opens application. 2. System displays username and password input fields, and has buttons such as "Login", "Forgot Password?", and "Register". 3. User clicks on register. 4. System displays fields for username, email, restaurant, restaurant location, password, and displays a submit button. 5. User enters information for their account setup fields and clicks submit. 6. System encrypts and sends information to authorization server to store credentials. 7. System sends user an email to confirm their email address. 8. User clicks on the confirmation link and finishes account registration process.
Exceptions:	1.0.E.1 Required information is not provided (at step 5) 1. System informs the user that the necessary fields need to be filled out. 2. User inputs the required information and clicks the sign-up button. 3. Return to step 6.
Includes:	None
Priority:	High

Frequency of Use:	Low Frequency: User only registers an account for a new restaurant or when using the application for the first time.
Business Rules:	TBD
Special Requirements:	1. User should create and enter credentials with alphanumeric characters.
Notes and Issues:	None

3.6 Use Case 6: Login

Use Case ID:	6
Use Case Name:	Login
Created By:	Yusuf Bahadur
Last Updated By:	Yusuf Bahadur
Date Created:	October 8, 2019
Date Last Updated:	October 8, 2019
Actors:	Users
Description:	The user enters in their username and password to be authenticated into the application.
Preconditions:	1. Application is connected to a network.
Postconditions:	1. A welcome animation alerts the user of successful login. 2. System displays the main application page.
Normal Flow:	1.0 Enter Account Credentials to Sign-In 1. User opens application. 2. System displays username and password input fields, and has buttons such as "Login", "Forgot Password?", and "Register". 3. User inputs username, password, and clicks submit. 4. System encrypts and sends the credentials to the authorization server. 5. System searches for a match between the user credentials and all account holder credentials. 6. System authorizes the user and access is granted. 7. System displays a welcome animation followed by the main application page.
Alternative Flows:	1.1 User does not have valid login credentials (branch after step 5)

	1. System does not grant the user access. 2. System displays "Invalid Username and/or Password" 1.2 User clicks forgot password (after step 2) 1. System displays fields to enter users "username" and a submit button. 2. User enters username and presses submit. 3. System sends the username to authorization server. 4. System searches for username. 5. System finds username and sends the associated email address a "reset password" email. 6. User clicks on "reset password" email and resets credentials through web application.
Exceptions:	1.0.E.1 Required information is not provided (at step 3) 1. System informs the user that the necessary fields need to be filled out. 2. User inputs the required information and clicks the login button. 3. Return to step 4.
Includes:	None
Priority:	High
Frequency of Use:	Low Frequency: User only logs in the first time they open the application or if they decided to log out
Special Requirements:	1. User should already exist.
Notes and Issues:	None

3.7 Use Case 7: Change Restaurant Location

Use Case ID:	7
Use Case Name:	Change Restaurant Location
Created By:	Yusuf Bahadur
Last Updated By:	Yusuf Bahadur
Date Created:	October 17, 2019
Date Last Updated:	October 17, 2019
Actors:	Users
Description:	The user changes to a new restaurant location on the application.
Preconditions:	1. Application is connected to a network.

Postconditions:	1. Application will display that it is set to the new restaurant location 2. Application will display information pertinent to the new restaurant location.
Normal Flow:	1.0 Enter New Restaurant Location. 1. User opens application for the first time. 2. System displays username and password input fields, and has buttons such as "Login", "Forgot Password?", and "Register". 3. User enters information and logs in. 4. System displays a prompt for users to change to a new restaurant location. 5. User enters information for their new restaurant location and clicks submit. 6. System saves user information.
Alternative Flows:	1.1 User is not opening up the application for the first time on a Clover device. 1. User navigates to settings and clicks on "Restaurant Information." 2. User clicks on "Change Restaurant Location." 3. System displays a prompt for users to optionally change to a new restaurant location. 4. User enters information for their new restaurant location and clicks submit. 5. System saves user information. 1.2 User wants to delete restaurant location. 1. User navigates to settings and clicks on "Restaurant Information." 2. User clicks on "Manage Existing Locations." 3. System displays a list of all existing restaurant locations. 4. User clicks "Delete" next to a location. 5. System removes restaurant location and saves information.
Exceptions:	1.0.E.1 Restaurant location is not accepted as a real address by the Maps API 1. System informs the user that the address does not exist. 2. User inputs the correct address information.

	3. Return to step 6.
Includes:	None
Priority:	High
Frequency of Use:	Low, User would only use this feature when opening up a new restaurant.
Special Requirements:	1. None.
Notes and Issues:	None

3.8 Use Case 8: Add Preferred Service Providers

Use Case ID:	8
Use Case Name:	Add Preferred Service Providers
Created By:	Karla Sunjara
Last Updated By:	Karla Sunjara
Date Created:	October 19, 2019
Date Last Updated:	October 19, 2019
Actors:	Users
Description:	Upon a user's first login, a user can add one or more preferred service providers.
Preconditions:	1. User logs into the application for the first time. 2. The user has the correct permissions to add the preference.
Postconditions:	1. The preferred service providers are saved into the user's settings. 2. The system stores the user's data in the database.
Normal Flow:	1.0 Add Preferred Service Providers 1. The user logs into the application for the first time. 2. The system prompts the user to set up their account by inputting their restaurant location and preferred service providers. 3. User inputs their restaurant location and preferred service providers. 4. User confirms their list of preferred service providers and clicks "Next". 5. User is brought to the home page.
Alternative Flows:	1.1 User's preferred service provider does not have a Roopairs account (branch after step 3)

	1. The system notifies the user that their preferred service provider does not have a Roopairs account and prompts the user to input the email address of the service provider for an invitation to join Roopairs to be sent out. 2. The user inputs the email address of their preferred service provider. 3. The system sends the service provider an invite to create a Roopairs account.
Exceptions:	1.0.E.1 Required information is not provided (at step 3) 1. System informs the user that the necessary fields need to be filled out. 2. User inputs the required information and clicks the send service request button. 3. Return to step 4.
Includes:	None
Priority:	High
Frequency of Use:	Low, user would only do this during their first login, afterwards they can edit their preferred service providers through their settings.
Special Requirements:	1. User should be able to edit their list of preferred service providers.
Assumptions:	Assume that users have preferred service providers.
Notes and Issues:	1. The user's preferred service provider may not have an email address.

3.9 Use Case 9: Edit Preferred Service Providers

Use Case ID:	9
Use Case Name:	Edit Preferred Service Providers
Created By:	Logan Lawson
Last Updated By:	Logan Lawson
Date Created:	October 21, 2019
Date Last Updated:	October 21, 2019
Actors:	Users
Description:	A user can edit list of preferred service providers through the setting page.
Preconditions:	1. User is logged into the application. 2. User is on the settings page.

Postconditions:	1. The list of preferred service providers is updated in the database and on the user interface.
Normal Flow:	1.0 Edit Preferred Service Providers 1. The user clicks on the option to edit preferred service providers. 2. The system displays the list of preferred service providers. 3. The user clicks add a service provider. 4. The system prompts user for service provider information. 5. The user inputs service provider information. 6. The system updates the user's settings with the user's input.
Alternative Flows:	1.1 User's preferred service provider does not have a Roopairs account (after step 5). 1. The system notifies the user that their preferred service provider does not have a Roopairs account and prompts the user to input the email address of the service provider for an invitation to join Roopairs to be sent out. 2. The user inputs the email address of their preferred service provider. 3. The system sends the service provider an invite to create a Roopairs account. 1.2 User wants to remove a preferred service provider from the list (at step 3). 1. The user selects a preferred service provider. 2. The system displays all information pertaining to the service provider and gives user option to remove provider. 3. The user clicks remove service provider. 4. The system removes and updates the list of preferred service provider in the user's settings.
Exceptions:	1.0.E.1 Required information is not provided (at step 5)

	1. System informs the user that the necessary fields need to be filled out. 2. User inputs the required information and clicks the add preferred service provider button. 3. Return to step 6.
Exceptions:	1.1.E.1 Required information is not provided (at step 2) 1. System informs the user that the necessary fields need to be filled out. 2. User inputs the required information and clicks the invite service provider button. 3. Return to step 3.
Includes:	None
Priority:	High
Frequency of Use:	Low, user would edit their preferred service providers if they made a mistake from the Login page or if they gained a new preferred service provider.
Business Rules:	TBD
Special Requirements:	1. User should be able to edit their list of preferred service providers.
Assumptions:	Assume that users have preferred service providers.
Notes and Issues:	1. The user's preferred service provider may not have an email address.

4 System Features

4.1 Create a Service Request

4.1.1 Description

User will be able to create a service request.

4.1.2 Stimulus/Response Sequences

Stimulus	**Response**
User clicks on the create service request button.	System redirects the user to the send a service request page.
User fills out the text fields with the required information.	System displays text inside the text fields.
User submits the service request to the system.	System updates and displays the request on the tracking page.

4.1.3 Functional Requirements

FR-1a	System shall be able to create a service request.
FR-1b	System shall allow the user to cancel a service request.
FR-1c	System shall allow the user to view the details of a service request.
FR-1d	System shall allow the user to select a flexible or urgent repair time frame.

4.2 Accept a Pending Service Request

4.2.1 Description

User will be able to accept a quote from a service provider after creating a service request.

4.2.2 Stimulus/Response Sequences

Stimulus	**Response**
User opens application.	System sends a service acceptance request pop-up.
User chooses a service provider from the list.	System displays the information for the service provider quote.
User accepts the service provider quote.	System updates and displays the request as scheduled on the tracking page.

4.2.3 Functional Requirements

FR-2	System shall allow the user to select a service provider for a service request.

4.3 Add Equipment Information

4.3.1 Description

User will be able to add an equipment to their list of equipment. Equipment information will include things like name, serial number, make, model, and location in the kitchen.

4.3.2 Stimulus/Response Sequences

Stimulus	**Response**
User clicks the button to add an equipment.	System redirects user to the adding equipment page.
User fills out the text fields with the required information.	System displays text inside the text fields.

4.3.3 Functional Requirements

FR-3a	System shall allow the user store details about the restaurant equipment.
FR-3b	System shall allow the user to view the details about restaurant equipment.
FR-3c	System shall allow the user to filter restaurant equipment based on category.
FR-3d	System shall allow the user to see past service requests for a piece of equipment.

4.4 View Past Service Requests

4.4.1 Description

User will be able to view an equipment's past work history. Past work history will include a list of previous service calls for each individual piece of equipment.

4.4.2 Stimulus/Response Sequences

Stimulus	Response
User clicks on the past service requests option of the restaurant equipment page.	System displays past service requests in a date ordered table structure.
User clicks on a past service request.	System displays full invoice for that service request.

4.4.3 Functional Requirements

FR-4	System shall allow the user to view all past service requests.

4.5 Register for a Roopairs account

4.5.1 Description

User will be able to register for a Roopairs account in order to access the application.

4.5.2 Stimulus/Response Sequences

Stimulus	Response
User fills out the text fields with the required information.	System displays text inside the text fields.
User clicks on the "sign-up" button.	System sends information to authentication server to add user.

4.5.3 Functional Requirements

FR-5	System shall allow the user to register a Roopairs account through the application,

4.6 Login

4.6.1 Description

User will be able to login to the application.

4.6.2 Stimulus/Response Sequences

Stimulus	Response
User fills out the text fields with the required information.	System displays text inside the text fields.
User clicks on the "sign-up" button.	System sends information to authentication server to add user.

4.6.3 Functional Requirements

FR-6	System shall allow the user to log in to the application.

4.7 Change Restaurant Location

4.7.1 Description

User will be able to change to new restaurant location.

4.7.2 Stimulus/Response Sequences

Stimulus	Response
User fills out the text fields with the new restaurant location information.	System displays text inside the text fields.
User clicks on the "submit" button.	Systems saves the information in the database and displays the information in the user's settings page.

4.7.3 Functional Requirements

FR-7	System shall allow the user to change the restaurant location.

4.8 Add Preferred Service Providers

4.8.1 Description

User will be able to add preferred service providers.

4.8.2 Stimulus/Response Sequences

Stimulus	Response
User logs into the Roopairs application for the first time.	System prompts the user to input their restaurant location and preferred service providers.
User fills out the text fields with their restaurant location and preferred service providers.	System displays text inside the text fields.
User clicks on the "save" button.	Systems saves the information in the database and displays the information in the user's settings page.

4.8.3 Functional Requirements

FR-8	System shall allow the user to create a list of preferred service providers.

4.9 Edit Preferred Service Providers

4.9.1 Description

User will be able to edit the list of preferred service providers.

4.9.2 Stimulus/Response Sequences

Stimulus	Response
User clicks on option to edit preferred service providers on the settings page.	The system displays the list of preferred service providers.
The user clicks add a service provider.	The system prompts user for service provider information.
The user inputs service provider information.	The system updates the user's settings with the user's input.

4.9.3 Functional Requirements

FR-9a	System shall allow the user to invite service providers to the Roopairs platform.
FR-9b	System shall allow the user to edit the list of preferred service providers.

5 External Interface Requirements

5.1 User Interfaces

UI Requirement	Description
UI-1	System shall have a screen that allows the user to log-in.
UI-2	System shall have an interface for viewing and managing all restaurant equipment.
UI-3	System shall be able to display work history for restaurant equipment.
UI-4	System shall have an interface to display available service providers.
UI-5	System shall render display sizes that appropriately fit different screen sizes.

5.2 Hardware Interfaces

HI Requirement	Description
HI-1	System shall run on custom Clover hardware.
HI-2	System shall be able to handle screen rotations for the Clover Station device.
HI-3	System will run on touch screen devices.

5.3 Software Interfaces

SI Requirement	Description
SI-1	System shall save user account credentials on the application.
SI-2	System shall run on Clover's hardened Android Software.
SI-3	System shall execute all service requests through the Roopairs API.

5.4 Communications Interfaces

CI Requirement	Description
CI-1	System shall transmit user and restaurant data to the server via an API.
CI-2	System shall transmit service requests through an API.
CI-3	System shall work with the API to transmit service requests to intended recipients.
CI-4	System shall notify users using notifications when they receive a service request update.

6 Other Nonfunctional Requirements

6.1 Performance Requirements

PR-1	System shall allow maintenance updates only from 1:00AM to 4:00AM to refrain from interfering with restaurant business hours.

6.2 Safety Requirements

SR-1	System shall preserve and save application data in the event that the network connection is broken mid-interaction.

6.3 Security Requirements

SCR-1	System shall not store payment data anywhere.
SCR-2	System shall handle payment data through an external payment platform, Stripe.

6.4 Software Quality Requirements

SQR-1	System shall poll the server for requests every 60 seconds.
SQR-2	System shall allow the user to create a service request in under 5 total clicks. 3 clicks max to get to the request form, 1 click to send the request, and 2 clicks max to confirm the service provider.

7 User Personas

7.0.1 John Stracciatella

Age: 47
John Stracciatella is a 47-year-old who runs a small family restaurant business. John just opened his third restaurant location. John has been in the restaurant business since he was 15 years old, when he started working as a waiter for Pedones, a local Italian restaurant. In his coming years, John became very passionate about cooking and decided take up culinary school. As his senior project he opened his first restaurant. Although the initial two years were rocky, it did not take long for his restaurant to flourish, becoming one of the local favorites.

Fast-forward twenty years, John now runs his restaurant from three different locations. However, if there is anything John loves more than cooking, it is technology. That is why John was one of the first restaurants to incorporate the Clover POS system into his business. It has saved him time and therefore money by helping him manage his sales, inventory, customers, and employees. However, one of the features that is missing is the management of equipment. With three different restaurants operating, John's biggest fear is when a kitchen appliance breaks down. Just last year, John lost $10,000 in one night because his oven broke down and he could not get a hold of a technician until later the next day.

In order to get a technician, John had to make 15 different calls to find a repairman that could fix his oven. As a tech enthusiast John wishes there were a way he could automate this whole process. He knows the capabilities of the Clover technology and wishes it could help him find a technician faster.

7.0.2 Paul Hobart

Age: 60

Paul Hobart is someone who has struggled to keep up with the times. It seems that every year technology is released with new features that he does not understand. Paul has a wife and two sons that help out with anything technology related, but he has fallen victim to a few phishing scams. It has become a frustration for Paul, as he prefers the older days when things were much simpler and not constantly changing. In his free time, he loves fishing, cooking something up for his family, and watching his favorite sports teams on television.

Although Paul has been in the restaurant business for all of his career after graduating from culinary school, he feels overwhelmed as the competition continuously innovates

with new solutions for common problems faced in the industry. He owns eight restaurants that have been top-rated for years. Even the Yelp reviews for his restaurants are stellar on the food side, but there are a few complaints when it comes to waiting for the food. When a piece of equipment breaks down, it spells disaster for the night as the cooks scramble to work around it.

Luckily, his two sons that also help out with the business have followed other restaurants and incorporated a POS system in recent years. However, Paul has struggled with this adjustment, since there are so many features that require company-wide training for working the software. His sons have been doing some research and figured out that there are some apps that can help save the businesses money, such as one that helps schedule repairs almost instantly. Paul is delighted to know that something like this exists, but is hesitant due to his lack of technology expertise. Luckily, this application does not require training at all and is mostly intuitive for the users with an ease of use.

7.0.3 Chelsea Tolhurst

Age: 19

Chelsea is a 19 year old student at Cal Poly San Luis Obispo. Chelsea is a 2nd year communications major and to get through school, she has to work part time at a restaurant in downtown San Luis Obispo. She has experience with technology through her laptop and her phone, but nothing too fancy because she's just a communications college student.

Chelsea has been working at the same restaurant for 2 years now. She started off as a hostess and now she's working as a waitress. She loves her job because she gets the opportunity to meet new people from all over the city. She also loves her coworkers because most of them are Cal Poly students like her so they all get along very well.

The restaurant she works for is old and some of the appliances and equipment are outdated. There are times during the lunch rush or the dinner rush when something will break down, like the ice machine or the soft drink machine. Chelsea gets really frustrated whenever something breaks down because she has to take the time out of her busy shift to find the manager on duty and explain what happened. Also, because the machines are out of order, Chelsea has to risk the chance of disappointing her customers because they might not be able to order what they would've liked.

Chelsea wishes there was a way for the equipment in the restaurant to be fixed quickly and efficiently. Luckily, another one of her friends works at Mama's Meatballs, which is another restaurant in downtown SLO. Her friend told her about the system they use at her work, Clover, which has an app available for download that can easily notify the

manager about an equipment breakdown and send out service requests to multiple service providers.

7.0.4 Roy Matthews

Age: 53

Roy Matthews is a 53 year old that has spent almost his entire adult life in the restaurant business. Roy went to school at Allen Hancock where he studied business. During school he found his love for restaurants and realized that when he graduated he wanted to manage and own his own restaurant. He eventually worked his way through multiple restaurants until he was a partial owner and manager of a fancy Italian Bistro in the downtown area of San Luis Obispo.

Roy values his business and relies on his restaurant to run at full capacity during normal operating hours. Several things can impact his business operating at 100 percent, with broken equipment being the number one reason. He is constantly worried that his kitchen equipment will break down during the busy hours of the day and that he will not be able to find service technicians quick enough to save his projected revenue for that day.

Roy is brutally technology impaired and hates when he has to deal with complicated modern technology to function in his day to day life. Roy unfortunately went through a month when several pieces of his kitchen broke down and his regular repair technicians were unable to help him in a timely manor. He lost out on a ton of revenue and is looking for a way to improve his experience with on demand repair services. One of his restaurant colleagues recommended Roopairs to help with his service needs. Roy is willing to try it to help his restaurant, but he is scared because he is not tech-savvy.

7.0.5 Sandeep Khan

Age: 65

Sandeep Khan is a 65 year old man from Silicon Valley who manages Anita's Indian Restaurant. Sandeep went to school at UC Berkeley where he studied Computer Science and had a deep passion for technology. At Berkeley, Sandeep started the Robotics Club and worked hard to build his first computer. After college, Sandeep was hired at Google in Mountain View, CA where he made millions off of Google Stocks. At the age of 60, bored with the tech industry and not willing to retire, Sandeep decided to join his family restaurant as General Manager.

The restaurant that Sandeep works for is fast paced and serves upwards of 1000 customers a day. As such, Sandeep's biggest worry is that an appliance will suddenly break

down during the operating hours of the restaurant. This has happened on several occasions in the past month, usually resulting in slow service, angry customers, and low table turnovers. Anita Indian Restaurant's customers are becoming increasingly disappointed as appliances increasingly break down in the kitchen.

Sandeep is tired of having to spend hours to find a service technician and decides to search for a solution. He finds Roopairs, an application that can find and book service requests directly within the restaurant's Clover POS system. He is easily able to add the Roopairs application via the Clover App app store and save his restaurant with a few clicks.

A Glossary

POS System: Abbreviation for Point of Sale system. The combination of software and hardware that allows businesses to handle essential tasks such as completing a sales transaction, inventory management, etc.

Clover: A cloud-based Android point of sale platform that designs customizable POS devices.

Service Provider: Any business that provides repair services for restaurant equipment or restaurant related utilities.

Preferred Service Provider: A service provider that a restaurant favors over other service providers.

Service Request: A request that is sent from the user to notify the service providers that their restaurant is in need of service.

Pending Service Request: A service request is considered pending when the request has been sent but no service provider has accepted the job.

Scheduled Service Request: A service request is considered scheduled when a service provider accepts the job, sends the user a quote for their service, and the user accepts the quote.

Past Service Request: A past service request is a service request that has already been completed.

API: Abbreviation for application programming interface. It is a communication protocol between a client and a server intended to simplify the building of client-side software.