Project Name: Roopairs
Software Requirements Specification
version 1.0

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

October 10, 2019

Contents

Credits

Name	Date	Role	Version
Luke Reckard	October 10, 2009	Document Owner, Co-Author of Software Requirements Specification	1.0
Karla Sunjara	October 10, 2009	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

1 Introduction

1.1 Purpose

This document presents the requirements and limitations for the Roopairs service request application 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. Other Nonfunctional Requirements
5. External Interface Requirements

1.3.2 Project Owner - Ray Bartolomucci

Ray, the team's connection with Roopairs, will use this document to ensure that the development team understands the key features of the end product. *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 new 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

1.5 References

2 Overall Description

2.1 Product Perspective

2.2 Product Features

FE-1	Customer will request service from repair services when the restaurant's equipment breaks down.
FE-2	Customer will log and keep track of their restaurant's equipment.
FE-3	Past jobs on a restaurant's equipment will be available for the customer to look back on.
FE-4	Customer can view the invoices of all past service jobs.
FE-5	Application will allow customers to keep track of their restaurant's kitchen layout.
FE-6	Customer will receive analysis from all their equipment and past services in order to help the customer make informed decisions in the future.
FE-7	Application will have different levels of permissions for staging service requests depending on a customer's position at the restaurant.

2.3 User Personas

2.3.1 John Stracciatella - Karla

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 platform 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 dollars 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.

2.3.2 Paul Hobart - Luke

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.

2.3.3 Chelsea Tolhurst - Jessica

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.

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.

2.3.4 Roy Matthews - Logan

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.

2.3.5 Sandeep Khan - Yusuf

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.

2.4 User Classes and Characteristics

2.5 Operating Environment

2.6 Design and Implementation Constraints

2.7 User Documentation

2.8 Assumptions and Dependencies

2.9 Functional Requirements

FR-1	User can log their restaurant equipment.
FR-2a	User can create a service request through the map layout.
FR-2b	User can click on a piece of equipment on the map to find out more information.
FR-3	User can create a service request.
FR-4	User can keep track of their restaurant equipment.
FR-5	User can login to application.

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 3, 2019
Actors:	Users
Description:	The user clicks on the add service request option and inputs all of the necessary data.
Preconditions:	1. User is logged into the application. 2. The user has the correct permissions to send the request.
Postconditions:	1. A confirmation email is sent to the restaurant.
Normal Flow:	1.0 Send a Service Request

	1. User clicks on the add service request option. 2. System displays input fields the user must fill in such as equipment type, location of appliance, etc. 3. User inputs data and clicks the send service request button. 4. System sends the request to the Roopairs service that will then notify repair companies. 5. When a repairman accepts the job and sends back a quote, the system sends the user a service approval request. 6. The user approves the service approval request 7. The system sends the user an e-mail confirming the order. 8. The system stores the service request in the database.
Alternative Flows:	1.1 User cancels service request (branch after step 12) 1. User cancels the service request. 2. The system notifies the repairman of the cancellation. 3. The system stores the cancellation in the database. 1.2 User wants to edit the service request before approval (after step 4) 1. User clicks on the edit service request. 2. User changes the necessary data fields and clicks the send service request button. 3. Return to step 4. 1.3 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 repair company. 3. Repair company updates any necessary information and accepts or rejects new request. 4. Return to step 5. 1.4 No technicians are available, every technician has denied the service request (after step 4)

	1. System notifies the user that no technician is currently available. 2. System continues to check for available technicians. 3. Return to step 5.
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
Business Rules:	TBD
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)
Assumptions:	Assume that a piece of equipment will break down and need servicing
Notes and Issues:	1. The default date is the current date for the request to be sent in.

3.2 Use Case 2: Send a Service Request through Map

Use Case ID:	2
Use Case Name:	User can send a service request through the map layout
Created By:	Luke Reckard
Last Updated By:	Luke Reckard
Date Created:	October 3, 2019
Date Last Updated:	October 3, 2019
Actors:	Users
Description:	The user clicks a layout button that reveals a map of the kitchen and chooses a piece of equipment to repair and send a service request.

Preconditions:	1. User is logged into the application. 2. The user has the correct permissions to send a request.
Postconditions:	1. A confirmation email is sent to the restaurant after sending a request.
Normal Flow:	1.0 Send a service request through the map layout 1. User clicks on the map option of the application. 2. System displays a map of the kitchen layout. 3. User clicks on the location of a piece of equipment that needs repairing. 4. System displays a screen to input a new service request, already having filled out the required equipment fields. 5. User inputs the required data and clicks the send service request button. 6. System sends the request to the Roopairs service that will then notify repair companies. 7. When a repairman accepts the job and sends back a quote, the system sends the user a service approval request. 8. The user approves the service approval request. 9. The system sends the user and the repairman a notification confirming the order. 10. The system stores the service request and relevant data in the database.
Alternative Flows:	1.1 User cancels service request (branch after step 9) 1. User cancels the service request. 2. The system notifies the repairman of the cancellation. 3. The system notifies the user of a successful cancellation. 4. The system stores the cancellation in the database. 1.2 User wants to edit the service request before approval (after step 6)

	1. User clicks on the edit service request option in the requests display. 2. User changes the necessary data fields and clicks the send service request button. 3. System updates the request in the scheduling system for the repair services. 4. Return to step 7. 1.3 User wants to edit the service request after repair approval (after step 6) 1. User clicks on the edit service request option in the requests display. 2. User changes the necessary data fields and clicks the send service request button. 3. System updates the request for the repair company. 4. Repair company updates any necessary information and accepts or rejects new request. 5. Return to step 7. 1.4 No technicians are available, every technician has denied the service request (after step 6) 1. System notifies the user that no technician is currently available. 2. System continues to check for available technicians. 3. Return to step 6.
Exceptions:	1.0.E.1 Required information not provided (at step 5) 1. System informs user that the necessary fields need to be filled out. 2a. User puts in required information. 2b. Return to step 5.
Includes:	None
Priority:	Low
Frequency of Use:	Medium
Business Rules:	TBD
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 = low)

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

3.3 Use Case 3: Inputting an Appliance's Information

Use Case ID:	3
Use Case Name:	User inputs an appliance's information
Created By:	Jessica Chang
Last Updated By:	Jessica Chang
Date Created:	October 7, 2019
Date Last Updated:	October 7, 2019
Actors:	Users
Description:	The user adds a new appliance to the application.
Preconditions:	1. User is logged into the application. 2. User is on the appliances page. 3. The user has the correct permissions to add appliances.
Postconditions:	1. User can view the new appliance's information through the application.
Normal Flow:	1.0 Add an appliance through the application 1. User clicks on adding appliance option of the application 2. System displays texts fields for user to fill out. 3. User inputs name of the appliance, serial number, make and model, and where it is located in the kitchen. 4. User clicks the save button. 5. System saves the appliance data in the database.
Alternative Flows:	1.1 User cancels adding an appliance (branch after step 3) 1. User clicks the cancel button. 2. The system returns to the appliance page. 1.2 User adds an appliance that's already been added (branch after step 4)

	1. The system notifies the user that the appliance has already been added. 2. The system returns to the add appliance 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.
Business Rules:	TBD
Special Requirements:	None
Assumptions:	Assume that the restaurant owners will want to input their appliances in case the equipment breaks down.
Notes and Issues:	None

3.4 Use Case 4: Login

Use Case ID:	4
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.
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 "Sign Up". 3. User inputs username, password, and clicks submit. 4. System sends the credentials to the authorization server. 5. System searches for a match between 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 credentials (branch after step 5) 1. System does not grant the user access. 2. System displays a "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 thru web application. 1.3 User wants to sign-up (after step 2)

	1. User clicks on sign-up. 2. System displays fields for username, email, and password, and also displays a button for submit. 3. User enters information for their account setup fields and clicks submit. 4. System sends information to authorization sever to store credentials. 5. System sends user a email to confirm email address. 6. User clicks on confirmation link and finishes account sign-up process.
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:	High Frequency: User logs in everytime they open the application.
Business Rules:	TBD
Special Requirements:	1. User should be create and enter credentials with alphanumeric characters.
Notes and Issues:	None

3.5 Use Case 5: Viewing past work history

Use Case ID:	5
Use Case Name:	Viewing past work history
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 work history for individual appliances.
Preconditions:	1. User is logged into the application. 2. User is on the appliances page.

Postconditions:	N/A
Normal Flow:	1.0 View past work history for appliance through application. 1. User clicks on the past work option of the application. 2. System displays past work history 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:	N/A
Exceptions:	N/A
Includes:	None
Priority:	High
Frequency of Use:	Medium frequency. Will most likely be used when an appliance breaks down and a manager is reviewing past service history of that appliance.
Business Rules:	TBD
Special Requirements:	None
Assumptions:	Assume that the restaurant owners/managers will want to keep track of what breaks on an appliance and how often that appliance is out of service.
Notes and Issues:	None

4 System Features

4.1 Add Appliance Information - Jessica

4.1.1 Description and Priority

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

4.1.2 Stimulus/Response Sequences

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

4.1.3 Functional Requirements

FR-1	User can log their restaurant equipment.

4.2 Use the Map Layout - Luke

4.2.1 Description and Priority

User will be able to view a map layout of their appliances and create a service request for a specific appliance. Appliance information will automatically be filled in based on the list of existing appliances.

4.2.2 Stimulus/Response Sequences

Stimulus	**Response**
User clicks on an appliance in the map.	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.

4.2.3 Functional Requirements

FR-2a	User can create a service request through the map layout
FR-2b	User can click on a piece of equipment on the map to find out more information

4.3 Create a Service Request - Karla

4.3.1 Description and Priority

User will be able to create a service request.

4.3.2 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.

4.3.3 Functional Requirements

FR-3	User can create a service request.

4.4 View past work history - Logan

4.4.1 Description and Priority

User will be able to view an appliance'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 the button to view an appliance.	System redirects user to the appliance page.
User selects the past work history tab.	System displays list of past work requests in a table structure.

4.4.3 Functional Requirements

FR-4	Users can keep track of their restaurant equipment.

4.5 Login - Yusuf

4.5.1 Description and Priority

User will be able to login to 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 "login" button.	Systems takes the users credentials and validates it against authentication server.
User clicks on the "sign-up" button.	System sends information to authentication server to add user.

4.5.3 Functional Requirements

FR-5	User can login to application.

5 External Interface Requirements

5.1 User Interfaces

5.2 Hardware Interfaces

5.3 Software Interfaces

5.4 Communications Interfaces

6 Other Nonfunctional Requirements

6.1 Performance Requirements

Luke: A nonfunctional requirement would be maintenance updates nightly for the application, so that it would be ready to use in the morning. This would be from 1:00 AM to around 4:00 AM so that the app can be constantly performing at its best for the users. Without constant updates, users could become frustrated with potential bugs and lack of fast service.

6.2 Safety Requirements

Logan: A nonfunctional requirement would be to preserve platform data in an event that network connection is down. The platform would save any service request or other out going data interactions until network connection is restored.

6.3 Security Requirements

Jessica: There is a possibility that the restaurant side customers will want to pay for the repair services through our application. They could also save their payment information on the application for ease of checkout in the future. Because of this, our application will need to keep the customer's private payment information secure. Any breach of privacy may result in lawsuits against our application.

6.4 Software Quality Attributes

Karla: The application must poll the server on request every minute. This allows the system to stay updated for its users, but also not overwork itself by updating in too short time spans.

Yusuf: A nonfunctional requirement would be a user can create a service request in

under 5 total clicks from the home page to a finalized request order. This would help to ensure the user can submit a request in a short amount of time.

7 Other Requirements

A Glossary

B Analysis Models

C Issues List