

SRS Specification

Retro Rabbits Claim System

Ryan Hartley

Oluwatosin Botti

Tlou Lebelo

Baven Pavaday

May 3, 2019

Contents

1	Introduction	3
2	Domain Model	4
3	User characteristics	5
4	Functional Requirements	6
4.1	Use cases	6
4.2	Requirements	8
4.2.1	Application	8
4.2.2	Database	8
4.2.3	GPS	8
4.2.4	Image Recognition	8
4.2.5	Log-in and Authentication	9
4.2.6	Client Accounts	9
5	Quality Attributes	10
6	Traceability Matrix	12

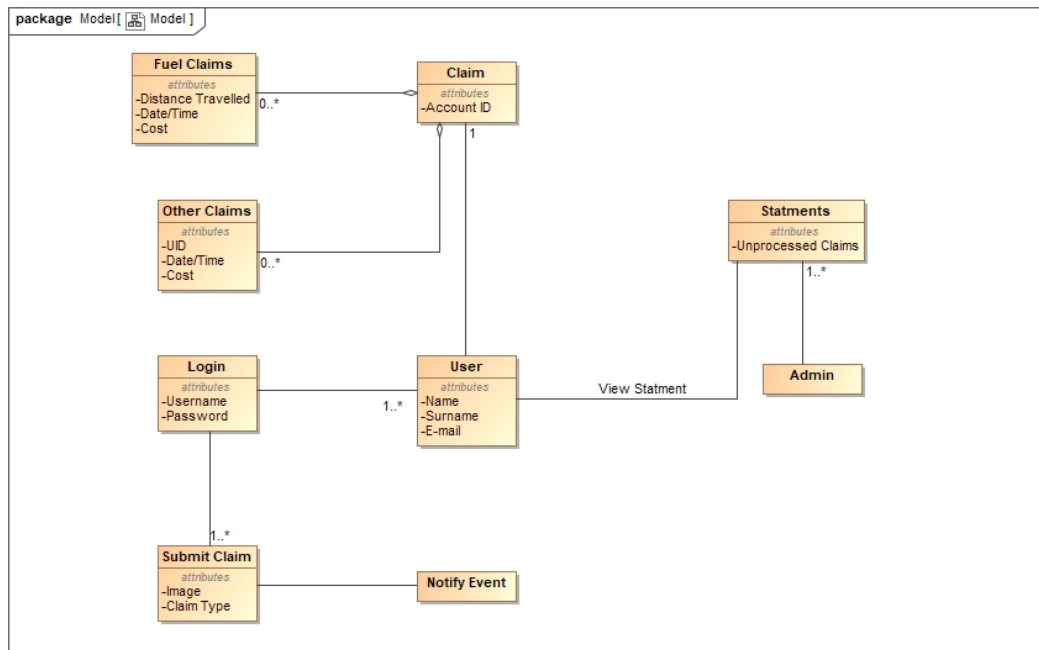
1 Introduction

Retro Rabbits current claims process is somewhat messy and a tedious process for the all the employees working at Retro Rabbit. At the moment, the employees are required to keep their slips from the past month (up to three months) which they are then allowed to claim for on the first Friday of the month. This administration process is difficult to manage and if an employee loses a slip, they aren't able to claim for it. In addition to this, it is difficult to make accurate claims for fuel used driving to clients (in the event that the employee needs to take a detour that hasn't been accounted for on the map software being used).

We have been tasked to create an android application that will improve this claims process, making it easier and more efficient for all the employees at Retro Rabbit. There are two main groups of claims, that being fuel claims and other(eg. food, data) claims. The android application therefore needs to make use of image analysis software which allows an employee to take a picture of the slip, this will then take the necessary data from it (date, time, amount, UID) and store the claim for the employee. In the event of an employee claiming for fuel, the employee is required to take a picture before they leave the office or their home of their cars odometer, then once they arrive at the client be reminded to take a picture of their odometer. The application then performs the required calculation and stores the data for the employee. On the first Friday of the month, the application will tell the employee if there are any pending claims and remind them to send their claims through.

On completion of this project we hope to provide Retro Rabbit with an application that meets their needs and provides a much needed streamlined and efficient claims process.

2 Domain Model



3 User characteristics

No technical skills or high level education required from the users. The application will prioritize usability to allow users to navigate throughout the application at ease.

The users of the system can broadly be categorized as follows:

- **Employee**

This are the primary users of the system.

The user must have an android smart-phone on which to download, install and use the application.

The user will record their monthly expenditure which the employer will later on compensate. This will include:

- Taking pictures of the cars' odometer before and after trips.
- Taking pictures of receipts as proof of payment.

- **Administrator**

This is the secondary user of the system.

This user will use the system to receive and view claims from employees through the system.

PDF statements containing claim information will be sent to the user every month.

4 Functional Requirements

4.1 Use cases

Figure 1: Authentication UC1

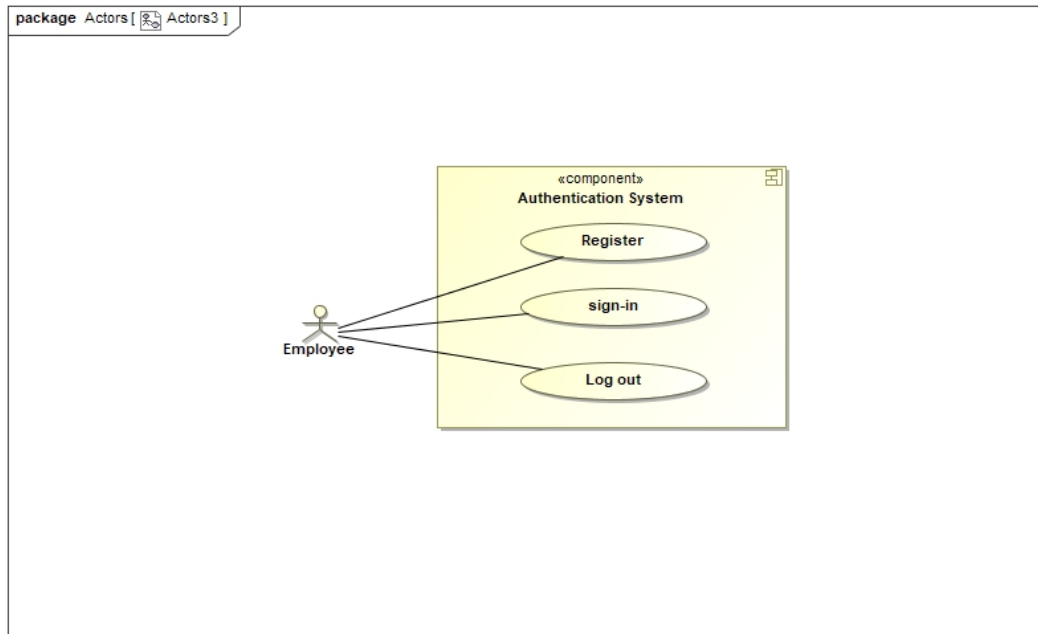


Figure 2: Other Claims UC2

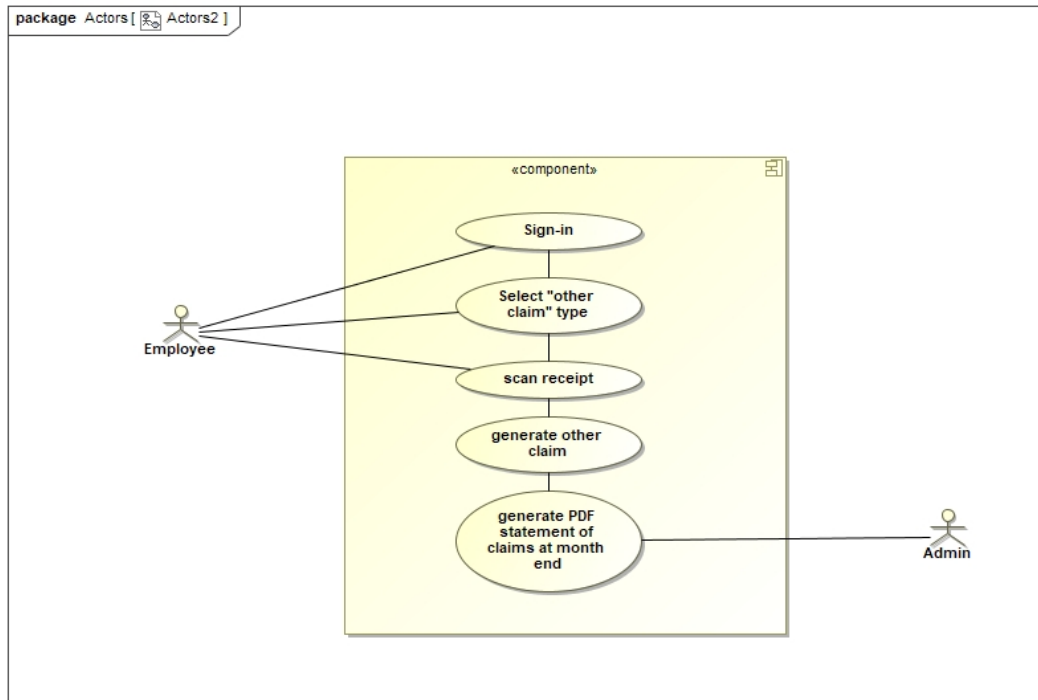


Figure 3: Unprocessed Claims UC3

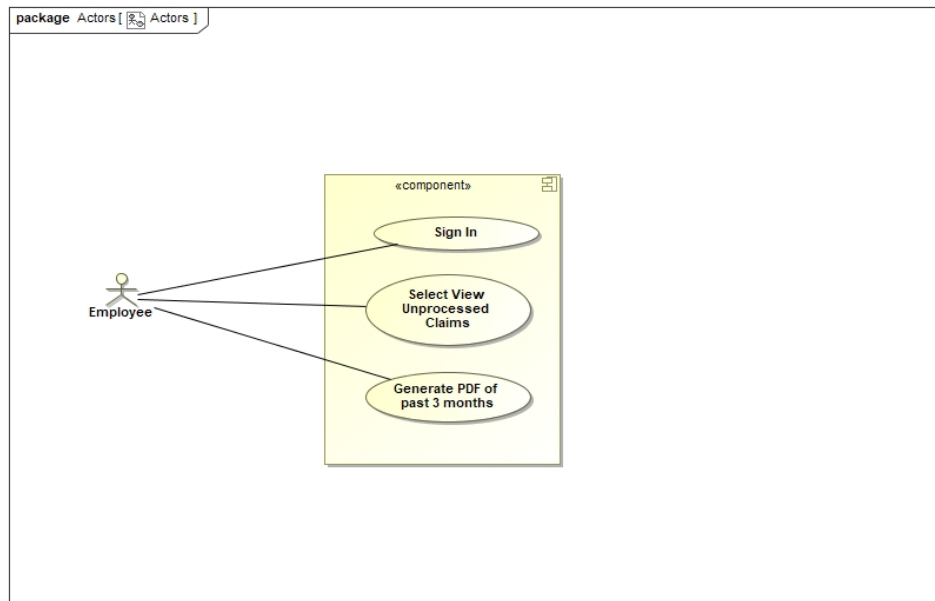
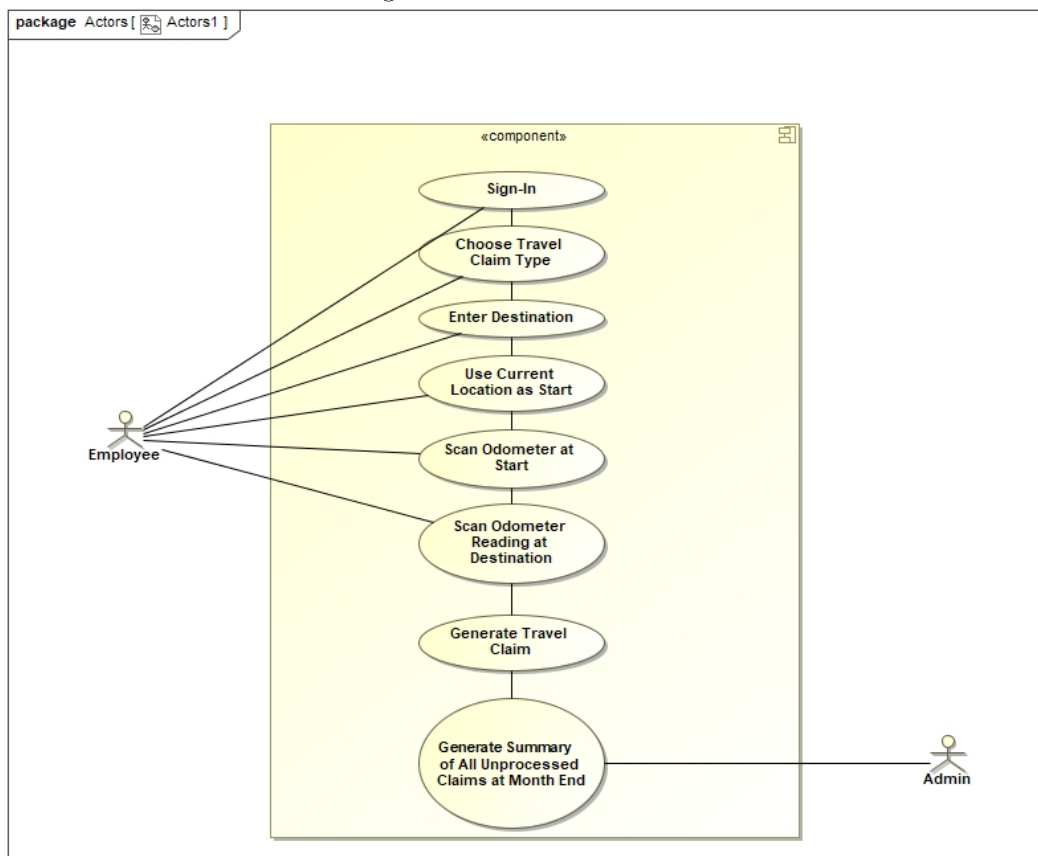


Figure 4: Travel Claims UC4



4.2 Requirements

4.2.1 Application

- R1.1 Provide a log in functionality for the employee.
- R1.2 Allow the employee to select the type of claim they would like to log.
- R1.3 Allow the employee to make use of their camera to scan the odometer or slip.
- R1.4 Allow the employee to view all unclaimed processes as well as the estimated cash back that can be expected(claim form).
- R1.5 Remind the employee via a push notification to log the odometer once they have arrived at their destination.
- R1.6 On the first Friday of the month remind the employee to log their claims for the month.

4.2.2 Database

- R1.7 Provide a NoSQL solution for data storage(compressed images and receipt/odometer data)
- R1.8 CRUD functionality for the data captured.
- R1.9 Statement generation using stored data.
- R1.10 Provide a suitable record structure for each claim.
 - 1. Store time/date
 - 2. ID
 - 3. The value(cost/ distance)

4.2.3 GPS

- R1.11 Get the employees current location from Google maps.
- R1.12 Calculates the the distance using the cars odometer between start and destination points.
- R1.13 Determine the shortest route for employee to take.
- R1.14 Compare the distance covered from the odometer reading to the shortest Google map route.

4.2.4 Image Recognition

- R1.15 Provide functionality to access the phone camera and take pictures.
- R1.16 Scan the images and provide a digital reading of:
 - 1. The odometers current value in kilometers.
 - 2. Date, time, unique ID an amount on the purchase slip.

4.2.5 Log-in and Authentication

- R1.17 The system should allow only valid employees to access the system.
- R1.18 The system should allow an employee to log in using some n-factor based authentication.
- R1.19 The system should allow an employee to log out.
- R1.20 The system should allow a new employee to be registered.

4.2.6 Client Accounts

- R1.21 The system should allow employees to view details on the claim form generated.
- R1.22 The system should allow an employee to edit details on the claim form generated.
- R1.23 The system should allow employees to view their unprocessed claims.

5 Quality Attributes

Performance:

- The system should be able to make timely(should take 2-3 minutes) claim forms that will be sent to the correct administrative powers handling the claims.
- The image to data recognition software used by the system should take less than a minute to process the information captured.
- Unprocessed claims within the last three months can be requested by the user at any point, this process should not take more than a minute.
- The application should provide the user with a timely(less than a minute) notification once a set destination has been reached in order to remind the user to take an odometer reading.

Scalability

- The system should allow for multiple concurrent users to be logged in and be able to process multiple requests to the server at the same time.
- The database solution will have to be highly scalable to allow for the potential influx of data therefore a NoSQL solution seems necessary.

Security:

- The system front-end will make use of at least 1 factor authentication system however the user will have a choice of verification techniques(using a password or authenticating through google).
- All local passwords used will be encrypted in a separate secure data storage.
- The data captured form the front-end will be encrypted and the encrypted data will be stored in the back-end.
- Extra security functionality will be added to the database used in order to prevent various data attacks.

Intergratability:

- All the defined subsystem just be able to communicate with each other using the API framework we will develop.
- External library functions that will be used for the image to data recognition must work within our system.
- The system must also be able to make use of Google services for both GPS services and potential authentication.

Testability:

- When uploading new code segments to our git we will make use of Travis CI to ensure that all the code submitted compiles.
- We will make use of test driven development for each of our subsystems to ensure that the system will be able to handle all possible corner cases of the problem.
- Our test functions will test our received output against the expected output for each possible case.
- We will make use of a multitude of unit testing tools(such as J-unit,Gradle etc.) to test each of the languages we will be using.
- There will be intergratability testing between not only our internal subsystems but also with the external modules used.

Usability:

- The system relies heavily on user interaction it is therefore imperative that the front-end sub-systems that the user interacts with is well designed
- The sub-systems that will be in constant interaction with the user must be designed so that the controls and functionalities are easy to understand and master.
- All controls will be well defined as well as navigation paths. At specific new segments of the application there will be an option to request help and tips in the form of helpful popups or notifications to assist the user in performing a certain functionality

6 Traceability Matrix

	Client Accounts	Application	Database	Image Recognition	Authentication	GPS
R1.1					X	
R1.2		X				
R1.3		X				
R1.4	X	X				
R1.5		X				X
R1.6		X			X	
R1.7			X			
R1.8			X			
R1.9		X		X		
R1.10			X			
1.11						X
1.12		X		X		
1.13						X
1.14				X		X
1.15		X				
1.16				X		
1.17					X	
1.18					X	
1.19					X	
1.20					X	
1.21	X	X				
1.22	X	X				
1.23	X	X				