

# **Requirements Analysis Document**

## **SKAJ Corp - Företagskvitton**

---

Version: 1.4

Date: 2017-04-05

Author: Sanjin Slavnic, Joakim Mattsson, Kevin Brunström & Anne Keller

This version overrides all previous versions

# Table of contents

<b>1. Introduction</b>	<b>3</b>
1.1. Definitions, acronyms and abbreviation	3
<b>2. Requirements</b>	<b>5</b>
2.1. User interface	5
2.2. Functional requirements	6
2.2.1. Overview	6
2.2.2. Table of application content	6
2.3. Non-functional requirements	7
2.3.1. Usability	7
2.3.2. Reliability	7
2.3.3. Performance	7
2.3.4. Supportability	7
2.3.5. Implementation	7
2.3.6. Packaging and installation	8
2.3.7. Legal	8
<b>3. Use cases</b>	<b>9</b>
3.1. Use case priority	9
<b>4. Domain model</b>	<b>10</b>
4.1 Class responsibilities	10
<b>5. References</b>	<b>11</b>
<b>6. Appendix</b>	<b>12</b>
6.1 Graphical User Interface (GUI)	12
6.2 Use-Cases	14
6.3 Domain-Model	44

# 1. Introduction

This application is sought after in the business industry, primarily to make life easier handling expenses. As of today there is no obviously easy way taking care of all your receipts. In order to do this you need to archive the receipts for and manually register all the data (expenses on the company) on a piece of paper each month. This is time consuming and inefficient.

Our goal is to develop an Android application that allows users to take a photo of a receipt and get all the data organised in the system directly. This application is wanted on the market by mostly traveling businessmen and women.

The application will be used every time a purchase has been made that needs to be registered. User should be able to effortlessly take a picture of their receipt and document it in the app, not being forced to bookkeep all expenses at once by the end of the month.

## 1.1. Definitions, acronyms and abbreviation

A.

Apache License

- The Apache License is a permissive free software license that allows the user of the software the freedom to use the software for any purpose.

B.

BottomBar

- A custom view component that mimics Material Design<sup>1</sup> Bottom Navigation pattern.

G.

Gradle

- Open source build automation system designed for multi-project builds.

GUI

- Graphical user interface.

J.

JSON

- JavaScript Object Notation is an language-independent open-standard format that uses text to transmit data objects consisting of serializable values.

---

<sup>1</sup> <https://material.io/>

L.

#### Library

- A library is a collection of resources, including pre-written code, classes, values, specifications and documentation used by programs to develop software.

M.

#### MVC

- Model-View-Controller is a way to organize the structure of an application. The Model containing the data and logic that determines how the application will operate. The View responsible for everything visible (the GUI) and Controller which handles user input and controls the model and view.

S.

#### Serializable

- The process of translating data structure or object state into a format that can be stored.

#### STAN

- Structure analysis tool for Java, visualizing code design, providing understanding of code, measuring quality and design flaws.

T.

#### TextRecognizer

- Object from Google library that collects text information from images.

X.

#### XML

- Extensible Mark-up Language is a format used for sharing data on the internet.

## 2. Requirements

### 2.1. User interface

We decided to have a Bottom Bar. It should provide five different options:

1. Browse in the receipt archive.
  - a. The archive should provide different ways to filter the list of all receipts. One should be able to filter list by dates, categories and price.
2. See graphical representation of statistics based on purchases.
  - a. The graphs tab should provide user with graphical representation of the data from receipts. Goal is to be able to do this by choosing three different diagrams which will give you the same information but be presented in three different ways.
3. Manage companies.
  - a. User should be able to manage his or hers companies, add or delete a company, add new employees and comment on all of the mentioned. The comments will be shown when you select an employee or a company.
4. Manage profile.
  - a. Users profile will display user's ID and photo and latest purchase. He or she should be able to edit their name and photo.
5. Go to the main page where you are able to create a new post.
  - a. The main page is where the user should be able to handle a new post (add a receipt). There should be an option to take a picture, continue without attaching a photo or go to your photo gallery to choose an already existing picture.

See appendix 6.1 for mock-ups of the GUI.

### 2.2. Functional requirements

#### 2.2.1. Overview

The user should be able to take a picture of a receipt, whose data then will be collected and shown in applications wizard guide. User will then input additional data or missing data the application couldn't collect from the picture. All information will be saved, available for the user to manipulate.

The user should also be able to get an overview of the receipts, visually shown in a list. Data will also be represented visually in different forms of chart and user should also be able to edit company's' information.

## 2.2.2. Table of application content

### 1. Add New Post

- a. Camera<sup>[CAMERA]</sup>
  - i. Select card<sup>2</sup>
    - 1. Company Card
      - a. Confirm company
      - b. Select supplier
      - c. Select/confirm date
      - d. Set/confirm total cost
      - e. Select category
      - f. Add comment
    - 2. Private Card
      - a. Select company
      - b. See above
- b. Gallery
  - i. See [CAMERA]
- c. No photo
  - i. See [CAMERA]

### 2. Archive

- a. List of receipts
  - i. Edit selected receipt
    - 1. Edit data
    - 2. Delete receipt

### 3. Company

- a. List of registered companies
  - i. Add new company
  - ii. Edit selected company
    - 1. Edit name
    - 2. Edit suppliers
    - 3. Delete company

### 4. Graphs

- a. Select type of graph
  - i. Receipt data represented graphically

### 5. Profile

- a. Edit name
- b. Edit photo
- c. Latest purchase

---

<sup>2</sup> This step is a requirement if the TextRecognizer is not able to collect the last 4 digits of a card registered on a company.

## 2.3. Non-functional requirements

### 2.3.1. Usability

The application will provide easy usability and run with Swedish as primary language. Wizard should lead user effortlessly through the process of adding a new receipt. No prior knowledge is needed for usage.

### 2.3.2. Reliability

Users will rely on the application's ability to guarantee all images are saved as they contain necessary data for end-of-month bookkeeping. Images of receipt will be saved locally on the mobile device providing the user to access them if needed from outside the application.

//TODO Ändrade detta om vi tillämpar Google Drive.

### 2.3.3. Performance

Users' interaction with the application should not result in noticeable stuttering or delays. There will be a loading bar when the application reads saved data. There will also be a loading bar when TextRecognizer is collecting data from an image. This is necessary to allow the application to load all accessible data.

### 2.3.4. Supportability

This application is supported by Android devices running Nougat 7.1 with API 25 or higher. It should be available for download on Google Play.

The source code will provide tests for the model. GUI should be tested manually while running the application.

### 2.3.5. Implementation

The application is written in Java using Android Studio IDE. Gradle build system will build the application in order to run it.

### 2.3.6. Packaging and installation

The application will run on in Android Studio and an Android device or emulator. See 2.3.4. Supportability for detailed information.

### 2.3.7. Legal

All libraries in use are distributed with the Apache 2.0 license. There will exist documentation in the code to inform what library are in use.

## 3. Use cases

See appendix 6.2 for Use Cases.

### 3.1. Use case priority

Below are Use Cases listed in order to priority.

High priority Use Cases:

1. Add new receipt
2. Archive

Medium priority Use Cases:

1. Company

Low priority Use Cases:

1. Graphical Representation
2. Profile



## 4. Domain model

See appendix 6.3 for Domain model.

### 4.1 Class responsibilities

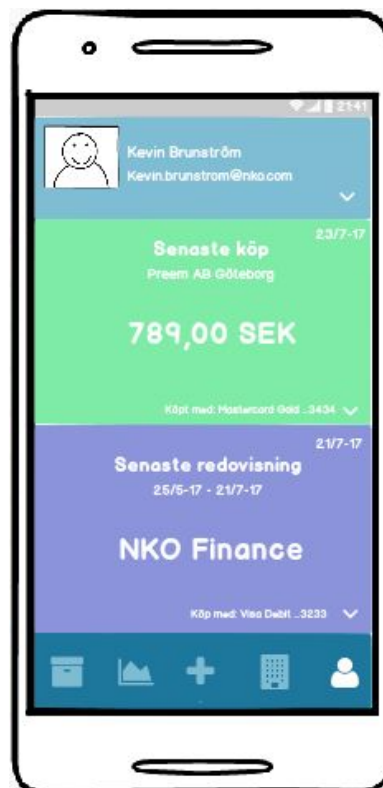
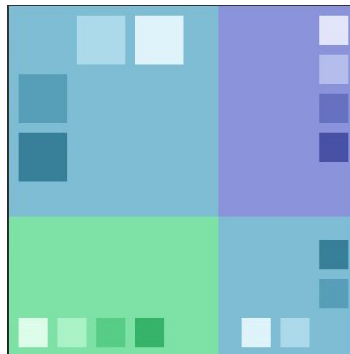
?

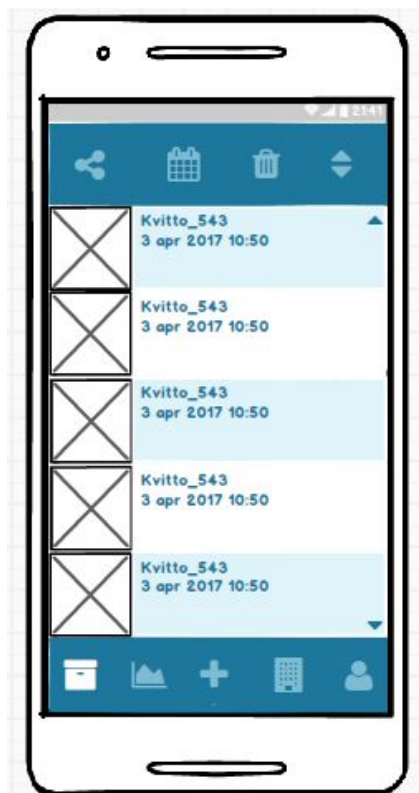
## 5. References

## 6. Appendix

### 6.1 Graphical User Interface

Below is colorscheme and mock-ups of our GUI.





## 6.2 Use-Cases

### UC: Add

**Summary:**

Register new receipt.

**Priority:**

High

**Extends:**

BottomNavigationView (App menu at the bottom of screen. Always visible)

**Includes:**

“Kamera”

“Galleri”

“Ingen bild”

**Participators:**

App user

Normal flow - "Företagskort" (No business card saved)		
	Actor	System
1	Press "Skapa ny"	
2		Camera opens. User takes a picture of receipt.
3		Gather info from receipt. Display option: "Privatkort" "Företagskort"
4	User selects "Företagskort"	
5		Display option: List of companies List of suppliers
6	User selects company (and supplier)	
7		Display options: "Splitta kvitto" "Fortsätt"
8	User selects "Fortsätt"	

9		Display: List of categories for receipt
10	User selects category for receipt.	
11		Display gathered info based on settings. Let user edit/add or remove data that will be saved in receipt.
12	User confirms info.	
13		Display confirmation.

Alternate flow - "Privat kort" (No business card saved)		
	Actor	System
1	Press "Skapa ny"	
2		Camera opens. User takes a picture of receipt.
3		Gather info from receipt. Display option: "Privatkort" "Företagskort"
4	User selects "Privatkort".	
5		Display: List of companies List of suppliers
6	User selects company (and supplier)	
		Display option: "Slitta kvitto" "Fortsätt"
7	User selects "Fortsätt"	
8		Display List of categories for receipt.



9	User selects category	
10		Display gathered info based on settings. Let user edit/add or remove data that will be saved in receipt.
11	User confirms info.	
12		Display confirmation.

Alternate flow - Business card saved		
	Actor	System
1	Press "Skapa ny"	
2		Camera opens. User takes a picture of receipt.
3		Gather info from receipt. Display: List of companies List of suppliers
5	User selects company (and supplier)	
6		Display option: "Slitta kvitto" "Fortsätt"
7	User selects "Fortsätt"	
8		Display List of categories for receipt.
9	User selects category	
10		Display gathered info based on settings. Let user edit/add or remove data that will be saved in receipt.

11	User confirms info.	
12		Display confirmation.

Alternate flow - "Splitta kvitto" (No business card saved)		
	Actor	System
1	Press "Skapa ny"	
2		Camera opens. User takes a picture of receipt.
3		Gather info from receipt. Display option: "Privat kort" "Företagskort"
4	User selects "Företagskort"	
5		Display: List of companies List of suppliers
6	User selects company (and supplier)	
7		Display option: "Splitta kvitto" "Fortsätt"

8	User selects "Splitta kvitto"	
9		Display: <ul style="list-style-type: none"> <li>• Available information from receipt</li> </ul>
10	User selects textbox containing purchased items.	
11		Display: <ul style="list-style-type: none"> <li>• Purchased items.</li> </ul>
12	User selects items to register.	
13		Display: <ul style="list-style-type: none"> <li>• List of categories</li> </ul>
14	User selects category for receipt.	
15		Display gathered info based on settings. Let user edit/add or remove data that will be saved in receipt.
16	User confirms info.	
17		Display option: Return to step 11 to continue "Splitta kvitto" process. Confirm and end process.
18		(If process ended) Display: Confirmation.

Alternate flow - "Splitta kvitto" (Business card saved)		
	Actor	System
1	Press "Skapa ny"	
2		Camera opens. User takes a picture of receipt.
3		Gather info from receipt. Display option: "Splitta kvitto" "Fortsätt"
4	User selects "Splitta kvitto"	
5		Display: Available information from receipt
6	User selects textbox containing purchased items.	
7		Display: Purchased items.
8	User selects items to register.	

9		Display: List of categories.
10	User selects category for receipt.	
11		Display gathered info based on settings. Let user edit/add or remove data that will be saved in receipt.
12	User confirms info.	
13		Display confirmation.

Exceptional flow – TODO		
	Actor	System
1		
2		
3		



## UC: Archive

### Summary:

Overview of all your registered receipts

### Priority:

High

### Extends:

BottomNavigationView (App menu at the bottom of screen. Always visible)

### Includes:

List of receipts

### Participators:

App user

Normal flow - Export PDF		
	Actor	System
1	Press archive	
2		Display:  List of receipts (sorted by available choice)
3	User selects export	

4		Display:  Available export options Email Download PDF SMS ...
5	User selects option	
6		System handles request

Alternate flow - Edit receipt		
	Actor	System
1	Press archive	
2		Display:  List of receipts (sorted by available choice)
3	User selects receipt	
4		Display:  Expanded receipt

5	User edits receipt	
6		System saves changes

Alternate flow - Delete receipt		
	Actor	System
1	Press archive	
2		Display:  List of receipts (sorted by available choice)
3	User selects receipt	
4		Display:  Expanded receipt Options how to manage receipt
5	User chooses to delete receipt	
6		Display:  "Are you sure you want to delete this receipt?"

7	User confirms	
8		Display: Receipt is deleted

Alternate flow – Search		
	Actor	System
1	Press archive	
2		Display: List of receipts (sorted by available choice)
3	User types in what to search for in the search bar	
4		Display: List of search result

Alternate flow - Delete list		
	Actor	System

1	Press archive	
2		Display: List of receipts (sorted by available choice)
3	User selects/marks what to delete	
4		Display: Selected receipts is shown
5	User chooses to delete selected	
6		Display: User is asked to confirm the deletion of receipts
7	User confirms	
8		Receipts/list is deleted

<b>Alternate flow – Sort</b>		
	<b>Actor</b>	<b>System</b>

1	Press archive	
2		Display: List of receipts (sorted by available choice)
3	User chooses what to sort the list by	
4		Display: The sorted list is shown

# UC: Company

**Summary:**

Shows the different companies you can interact with.

**Priority:**

Medium

**Extends:**

BottomNavigationView (App menu at the bottom of screen. Always visible)

**Includes:**

All companies you are a part of.

**Participators:**

App user

Normal flow		
	Actor	System
1	Press Company	
2		Display: List of companies.
3	User selects to add a new company	
4		Display: Text fields to be filled in is shown: Company name Company comment Supplier Supplier comment Cards Employees Employee Comment  User is asked to confirm all details
5	User confirms details.	
6		Display: Confirmation.



7	User chooses to edit an existing company.	
8		Display: Options to edit is shown: Company name Company comment Supplier Supplier comment Cards Employees Employee Comment
9	User edits fields.	
10		Display: User is asked to confirm changes.
11	User confirms details.	
12		Display: Confirmation. Details are saved.

Alternate flow - Delete company		
	Actor	System
1	Press company	

2		Display: List of companies.
3	User selects company	
4		Display: Company information Delete company button
5	User deletes company	
6		Display: User is asked to confirm the action
7	User confirms	
8		Display: Company is deleted

Alternate flow - Comment about company		
	Actor	System
1	Press company	

2		Display: List of companies
3	User selects company	
4		Display: Company information Delete company button
5	User add/edits comment about company	
6		Display: User is asked to save the comment
7	User selects save	
8		Display: Confirmation

Exceptional flow - Delete the only existing company		
	Actor	System
1	Press company	

2		Display: List of companies
3	User selects company	
4		Display: Company information Delete company button
5	User presses delete company	
6		Display: Pop-up is shown, "You can not delete your only company. Company will not be deleted."

Exceptional flow - Delete the only existing employee		
	Actor	System
1	Press company	
2		Display: List of companies

3	User chooses company	
4		Display: Company information Delete company button Delete employee button
	User presses delete employee	
		Display: Pop-up, "You can not delete your only employee. Employee will not be deleted."

# UC: Graphical Representation

**Summary:**

Shows the list of receipts represented graphically

**Priority:**

Low

**Extends:**

BottomNavigationView (App menu at the bottom of screen. Always visible)

**Includes:**

List of receipts

**Participators:**

App user

Normal flow		
	Actor	System
1	Press Graph	
2		Display: The option to choose between three types of graphs
3	User seletcs type of graph	
4		Display: The graph is displayed as per users choice

# UC: Profile

**Summary:**

Shows the information about the active employee

**Priority:**

Low

**Extends:**

BottomNavigationView (Appmenu at the bottom of screen. Always visible)

**Includes:**

The user

**Participators:**

App user



Normal flow		
	Actor	System
1	Press Profile	
2		Display: Profile for the employee

Alternate flow		
	Actor	System
1		
2		
3		
4		

Exceptional flow		
	Actor	System
1		
2		
3		
4		

## 6.3 Domain-Model

