December 1, 2015 v1.11

Team C

PostCardBuddy

System Requirements

Authors of this document:
Emma Albertz
Caroline Brandberg
Linnéa Claesson
Billy Johansson
Johan Ju
Jacob Mejvik
Carl Rynegardh

Contents

1 Introduction

This document is written within the context of the course Requirements Engineering at Lund Institute of Technology, which the authors are currently enrolled in. They have been provided with a project mission from another group, specifying a product they want to see developed. This group has also acted as the key customer. The intention of this document is to specify the requirements of this product, namely PostCardBuddy.

2 Background

Everybody likes receiving postcards, but the process of sending them is tedious and takes too much effort. This is what PostCardBuddy hopes to change. PostCardBuddy is a mobile application that will simplify the process, whether you want to be creative and design your own postcards or make it easy for yourself and use a template postcard based on your location and send it to everyone in your contact list.

The application is perfect for every occasion you want to send a postcard. Grandma's birthday is coming up? Send a postcard of you and your cousins! Christmas is around the corner? Send everyone in your contact list a postcard of your cats! Away on vacation? Why not send a ready-made postcard that shows off the amazing beach to everyone in the office? Nobody needs to know it rained all week.

PostCardBuddy is the perfect tool when you want to let someone know you are thinking of them, no matter the occasion.

3 Definitions and terms

Device Mobile device on which it is possible to download an use applications.

Standard library Library of pre-existing images in application.

Phone gallery User's existing image gallery on phone.

Mobile user Person who owns a smart phone.

Payment solution A feature that makes it possible to charge the costumer in the application.

Payment service A company who provides a payment solution for applications.

Personalized postcards Postcards were the design is chosen by the person who sends the postcard.

Product The application described in this Requirement Specification.

Postal service A company that delivers mail to private citizens.

Recipient The person which a postcard is addressed to.

Printer of postcards The company who delivers the postcards from the printer to the postal service. In this project the key customer.

Supplier of images The companies or persons who supply the application with images for the standard library.

System The application described in this Requirement Specification.

4 System Requirements

4.1 Goal

The product aims to establish the key customer in the postcard sending market and shall achieve this through the following goals:

- Simplify the process of sending postcards
- Enable user to send personalized postcards
- It shall be possible to generate revenue through the system

4.2 Domain

Write short text about contents of this section.

4.2.1 Context Diagrams

The context diagram of the product can be found in figure 1. This diagram shows the interface that the application PostCardBuddy will interact with and the stakeholders who will interact with the application.

There are two stakeholders that will interact directly with the application; the mobile user and the supplier of images. The mobile user is the one that will use the application for creating its personalized postcard. For the front of the postcard the user shall be able to select an image from the applications standard library. These images will be delivered from supplier, which are shown as *Supplier of images* in the context diagram.

The application needs various of functionality. Some of these functionalities will be used from the users mobile phone. The camera available from the users mobile phone will be used to enable the user to take a picture. The GPS location of the user shall also be provided from the phone. The GPS location will be used to select which pictures will be presented first from the standard library. The contacts available in the users phone will also be used to select recipients. These three permissions shall be confirmed by the user. The user shall also be able to send a digital postcard, via email.

The user shall also be able to send a physical postcard. This will cost money, which will be taken care of by a payment service. When the payment is done there will be a franking of the postcard. Thereafter the postcard will be sent to a printer. The printer will be placed so that a supplier, shown as *Printer of postcards* in figure 1, easily can pic up the card and deliver it to the postal service. The last stage in this chain is the stage where the postal service deliver the physical post card to the desired recipient.

4.2.2 Stakeholders

Some of the stakeholders for PostCardBuddy are presented in Table 1. For each stakeholder there is a number to visualize how much prioritization each stakeholder have. The scale is from 1-5 where 1 represent a high priority and 5 a low priority.

Mobile user was given the highest priority since they will be the ultimate users of the product and without them there will be no market.

Printer of postcards will act as the key customer, since they placed the order of the product, and thereby given a high priority.

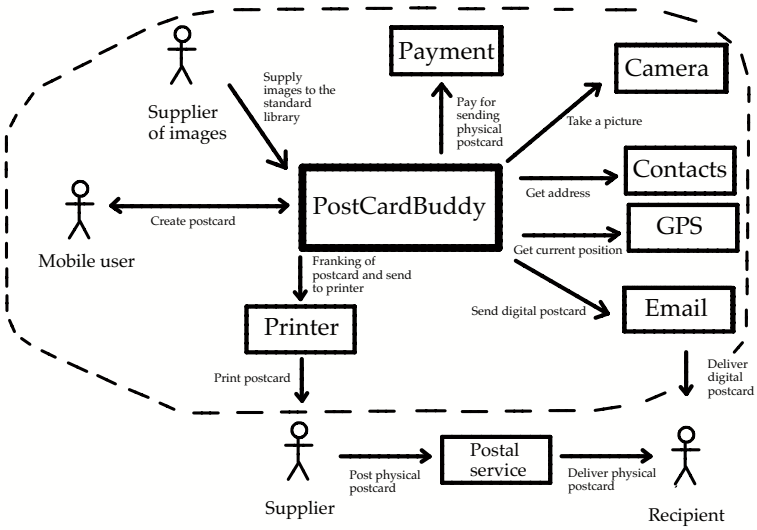

Figure 1: Context diagram of product.

Stakeholder	**Priority**
Mobile user	1
Printer of images	2
Postal service	2
Payment service	4
The existing application *Riktiga Vykort*	4
Developers	3

Table 1: Stakeholder prioritization

Postal service was given a high priority because they are considered a possible buyer of the application.

Payment service was given a low priority. The payment service will be used only to provide the application with a pay-functionality, and therefore its low priority.

The existing application *Riktiga Vykort* is a competitor to the application.

Developers decide whether a functionality is reasonable or not.

4.2.3 Tasks

Work area: Vacation

Communicating with friends and family. Usually from a remote location with unreliable internet access. Typically sunny and overall poor working environment. Simplicity is key to capturing important moments.

Users: Average smart phone user, used to little manual work.

Req 1.2.1.1 The system shall support tasks 1.1 and 1.2.

Task 1.1 Send a postcard.

Purpose: Take a picture. Edit the picture. Add a message. Add recipients. Send postcard.

Precondition: PostcardBuddy is running.

Sub-tasks:

1. Use the camera in the device to take a picture.
2. Allow basic editing of pictures.
3. Save pictures.
4. Add recipients from address book.
5. Save finished postcard.
6. Preview postcard.
7. Send postcard to printer.

Variants:

1a The user selects a picture from the personal gallery.
1b The user selects a picture from PostCardBuddy's standard library.
2a Include editing of library pictures.
4a Manually add address.

Task 1.2 History

Purpose: View sent cards. View recipients. View cost.

Precondition: PostcardBuddy is running.

Sub-tasks:

1. List sent postcards.
2. Search sent postcards.
3. Summary of cost.

Variants:

1a No postcards sent.
2a No postcards found for given search criteria.

4.2.4 Interfaces

Req 1.2.2.1 Printer The interface connecting PostCardBuddy and the printed postcards is an off-the-shelf printer.

Req 1.2.2.2 Print files The system sends image files to a printer.

4.3 Product

Short text describing this section.

Req 1.3.1.1 Success notification The user shall be notified when an order is sent from a device.

Req 1.3.1.2 Fail notification The user shall be notified when an order fails to be sent from a device.

Req 1.3.1.3 No internet If the user places an order on a device that is not connected to the internet, the order shall be stored and sent the next time the device receives internet connection.

4.4 Design

Short text describing this section.

Req 1.4.1.1 Front page The front of the postcard shall be a field containing an image.

Req 1.4.1.2 Text field The back of the postcard shall contain a text field.

Req 1.4.1.3 Address field The back of the postcard shall contain an address field.

Req 1.4.1.4 Postage field The back of the postcard shall contain a postage field.

Req 1.4.1.5 Postage print The postage shall be printed in the top right corner on the back of the postcard.

Req 4.4.0.6 Start Screen The application shall start with a screen where its possible to choose front and back image/text figure 2.

Req 4.4.0.7 Get image The application shall let the user choose the image source from a menu figure 3.

Req 4.4.0.8 Edit image The application shall give the user a basic image editor to customize the image figure 4.

Req 4.4.0.9 Recipiant address The application shall have an address input screen with an address-book/contacts (not in image) figure 5.

4.5 Data Requirements

Short text describing this section.

Req 1.5.1.1 Data model The system shall handle the data presented in the data model in figure 6.

4.5.1 Data dictionary

Class: Device
The device is the actual physical mobile device on which the application is running.

Examples:

1. An android device running the application.
2. An iOS device running the application

Attributes:

Figure 2: Start

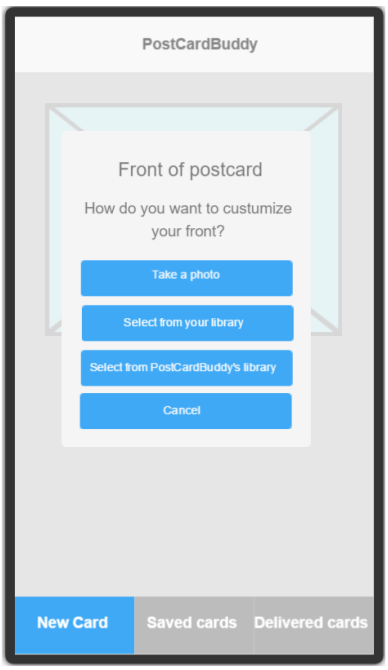

Figure 3: Chosee image source

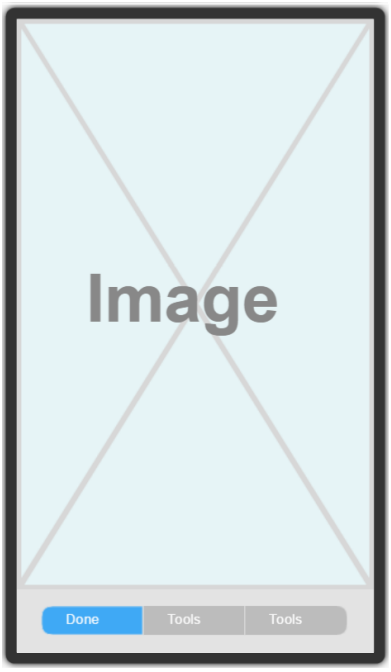

Figure 4: Image editor

Figure 5: Recipiant information

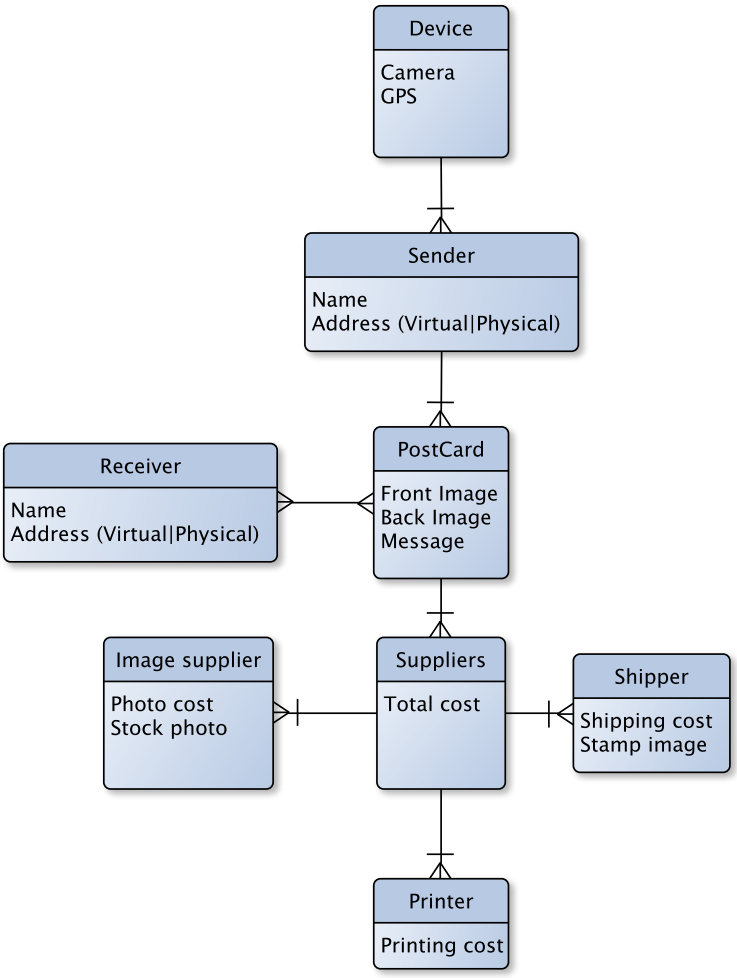

Figure 6: Data model

1. **Camera:** Image
   A compressed image fetched from the devices physical camera.
2. **GPS:** String[Latitude,longitude]
   The information about current coordinates from the GPS in the device. The string is given on the format shown and the latitude and longitudes are signed floats with seven decimal places.

Class: Sender

This class represents the person sending the post card. It can be the same person as the one using the device but it doesn't have to.

Examples:

1. The device owner.
2. A person using the application to send a post card.

Attributes:

1. **Name:** String
   The name of the sender.
2. **address** (Virtual—Physical): String
   This attribute is always a string. If it's a virtual address it's a user name, otherwise it's a physical street address.

Class: Receiver

This class represents the person receiving the post card. This class is identical to *Sender* in terms of attribute structure. The sender and receiver could be the same person.

Examples:

1. The person receiving the post card.
2. The same person as the one sending a post card.

Class: PostCard

This class represents the the post card sent from the *Sender* to *Receiver*. It encapsulates all the information necessary to send a post card in either virtual or physical form. An instance of this object, owning a *Sender* and a *Receiver* needs to exist to be able to send a Post Card.

Examples:

1. A post card with two images, a message and a stamp.
2. A post card with no images, no message and a stamp.
3. A virtual post card with images, a message and no stamp.

Attributes:

1. **Front image:** Image [optional]
   A compressed image that will be used as the front image of the *PostCard*.
2. **Back image:** Image [optional]
   A compressed image that will be used as the back image of the *PostCard*.
3. **Message:** String [optional]
   The message on the *PostCard*.
4. **Stamp image:** Image
   The image supplied by *Shipper* to properly send the post card.

Class: Suppliers

This class collects the data from an *Image supplier*, a *Shipper* and a *Printer*.

Examples:

1. A collection of suppliers relevant to printing and sending a specific *PostCard*.
2. Only a printing and shipping cost.

Attributes:

1. **Total cost:** Float
   The combined cost of *Image supplier/Photo cost*, a *Shipper/Shipping cost* and a *Printer/Printing cost*. The value is rounded up to two decimal places.

Class: Image supplier:
This class represents a supplier of images. If a user chooses a stock photo as (for example) a *PostCard/Front image* there is a cost with using the photo that needs to be added to the total cost.

Examples:

1. An image supplier with a photo and a cost.
2. An image supplier with a free photo.

Attributes:

1. **Photo cost:** Float
   The cost of a stock photo. The value is rounded up to two decimal places.
2. **Stock photo:** Image
   The actual image that will be bought.

Class: Shipper:
This class represents a shipper. The shipper is the company that will transport the post card.

Examples:

1. A representation of what is required to send a post card with Posten.
2. A representation of what is required to send a post card with DHL.

Attributes:

1. **Shipping cost:** Float
   The cost of shipping. The value is rounded up to two decimal places.
2. **Stock photo:** Image
   This is the image used on the post card to indicate that shipping was payed for.

Class: Printer:
This class represents a printer. The printer is responsible for printing the physical post card.

Examples:

1. A company contracted to print a post card.
2. The company supplying the application.

Attributes:

1. **Printing cost:** Float
   The cost of printing. The value is rounded up to two decimal places.

4.5.2 Virtual windows

Req 1.5.1.2 PostCard The input data to the *PostCard* class described in the Data dictionary shall include the items specified in the virtual window in figure 7.

Req 1.5.1.3 Sender The input data to the *Sender* class described in the Data dictionary shall include the items specified in the virtual window in figure 8.

Req 1.5.1.3 Sender The input data to the *Receiver* class described in the Data dictionary shall include the items specified in the virtual window in figure 9.

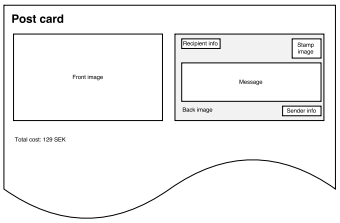

Figure 7: Virtual window PostCard

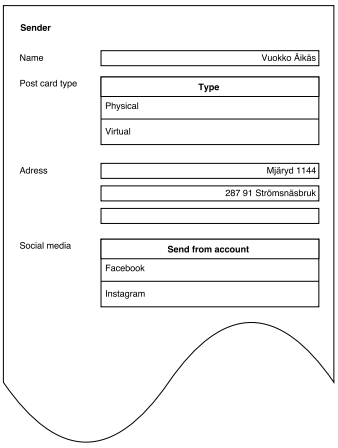

Figure 8: Virtual window Sender

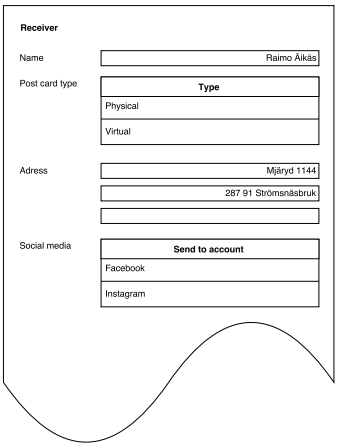

Figure 9: Virtual window Receiver

4.6 Functional Requirements

4.6.1 Images

Req 4.6.1.1 Image from phone gallery It shall be possible to choose pictures from the phone gallery for the front of the postcard.

Req 4.6.1.2 Picture from camera It shall be possible to take a picture through the camera and use as image for the front of the postcard.

Req 4.6.1.3 Image from standard library It shall be possible to choose pictures from a standard library.

Req 4.6.1.4 Image and GPS position Images from the standard library shall be presented based on the user's GPS position.

Req 4.6.1.5 Image editing The system shall have a function for editing of images.

Req 4.6.1.6 Image saving It shall be possible to save images.

4.6.2 Greetings

Req 4.6.2.1 Greetings It shall be possible to write greetings in the app.

Req 4.6.2.2 Pictures of handwritten greetings It shall be possible to choose a picture of a hand-written greeting.

Req 4.6.2.3 Auto-generated greetings It shall be possible to choose a template greeting.

Req 4.6.2.4 GPS based greetings The system shall be able to generate greetings based on GPS position.

Req 4.6.2.5 Handwritten greetings on screen It shall be possible to write a handwritten greetings directly on the screen.

Req 4.6.2.6 Saving greetings It shall be possible to save a greeting.

4.6.3 Recipients

Req 4.6.3.1 Enter recipients It shall be possible to enter recipients manually.

Req 4.6.3.2 Phone book recipients It shall be possible to choose recipients through the phone book.

Req 4.6.3.3 Multiple recipients The system shall be able to handle multiple recipients for one postcard.

Req 4.6.3.4 Favourite recipients It shall be possible to save recipients as favourites.

Req 4.6.3.5 Frequent recipients The system shall show frequently used recipients as favourite recipients.

4.6.4 Postcard

Req 4.6.4.1 Saving postcards It shall be possible to save postcards.

Req 4.6.4.2 Reuse postcards It shall be possible to reuse saved postcards.

Req 4.6.4.3 Preview postcards It shall be possible to preview postcards before sending it.

Req 4.6.4.4 Digital postcard It shall be possible to send digital postcards.

Req 4.6.4.5 Physical postcards It shall be possible to send physical postcards.

Req 4.6.4.6 Payment It shall be possible to pay for sending physical postcards.

Req 4.6.4.7 Postcard size It shall be possible to choose the size of the physical postcard.

Req 4.6.4.8 Quality of physical postcard It shall be possible to choose the print quality of physical postcards.

Req 4.6.4.9 History It shall be possible to display the history of sent postcards.

Req 1.6.3.8 Social media Feature for sharing postcards on social media.

4.7 Quality Requirements

4.7.1 Quality grid

Text about quality grid, reference table 2

Table 2: Quality grid

Quality factors - PostCardBuddy	**Critical**	**Important**	**As usual**	**Unimportant**	**Ignore**
Operation					
Integrity/Security			1		
Reliability/availability		2			
Usability	3				
Internet connection demand	4				
Efficiency			x		
Miscellaneous					
Installability		5			
Interoperability		6			

1. Text about item 1
2. 2
3. 3 etc.

4.7.2 QUPER

Text about quper diagram, reference figure 10.

4.7.3 Performance

Req 4.7.3.1 Memory usage The application shall adjust its memory usage depending on the device.

Req 4.7.3.2 Speed The user interface shall respond within 200ms after a finished user interaction if there should be an response. This is only applied on devices faster than Nexus 5 / iPhone 5 running original firmware without any other user installed applications on the phone.

Req 4.7.3.3 Picture quality The camera shall be able to take a picture in the highest hardware supported resolution.

Req 4.7.3.4 Autofocus The camera shall have a autofocus that is comparable to the Android / iOS stock camera.

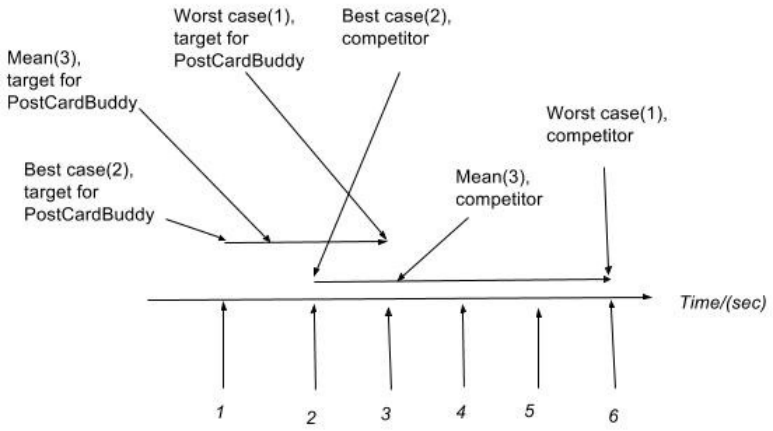

Figure 10: Text about the image here.

4.7.4 Availability

4.7.5 Security

Req 4.7.5.1 Store cards The photos shall be stored encrypted

4.7.6 Maintainability/Portability

Req 4.7.6.1 Language The application shall be developed in non native language e.g. Java for Android.

Req 4.7.6.2 Device support The application shall work on devices with newer operating systems than Android 4.1 / iOS 7.0.1

4.7.7 Usability

Req 4.7.7.1 User friendly 9 out of 10 users shall be able to use the system after a five minute instruction.

5 Release Plan