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

The process of sending postcards is time consuming and somewhat tedious. The product described in this document aims to solve this problem by providing an easy and efficent way to send personalized postcards directly from a mobile phone. Simply take a photo, choose a picture from the phones gallery or use a template picture. The template pictures can be found in the applications standard gallery and are based on the location of the user. To complete the postcard add a greeting and recipients. When the user presses the send button in the application the postcard is sent to a printer. The postcard is then delivered to the postal service for further forwarding to the final recipient.

3 Definitions and terms

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

Mobile user Person who owns a device.

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

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

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

Phone gallery User's existing image gallery on phone.

Postal service A company that delivers mail to private citizens.

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

Product The application described in this Requirement Specification.

Recipient The person which a postcard is addressed to.

Standard gallery Gallery of pre-existing images in application.

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
- Enable revenue generation

4.2 Functional Requirements

This section describes the functions and features of the application.

4.2.1 Domain

The domain level requirements provides information about how PostCardBuddy interacts with its surroundings.

4.2.1.1 Context Diagram

The context diagram of the product can be found in figure 1. This diagram shows the enivronment of the application and the stakeholders interacting with it.

There are two stakeholders that will interact directly with the application; the mobile user and the supplier of images. The mobile user will use the application for creating personalized postcards. For the front of the postcard the user shall be able to select an image from the applications standard gallery. These images will be delivered by a *Supplier of images*.

Some of the features of the application will require the use of functionality allready present in the the device. The camera will be used to enable the user to take a picture. The GPS will provide the location of the user. The contacts will be used to select recipients. Finally the E-mail functionality in the device will be used when sending electronic postcards.

The user will be charged through a standard payment solution integrated in the application. The charge includes both the cost of the postcard and the franking.

When a user sends a postcard it will be sent to the *Printer of postcards* who delivers the postcard to the postal service. Finally the postal service delivers the physical post card to the final recipients.

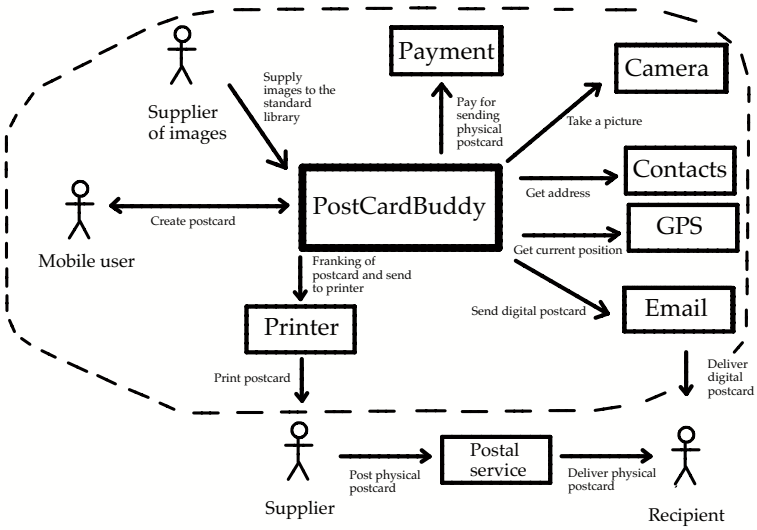

Figure 1: Context diagram of product.

4.2.1.2 Stakeholders

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

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

Table 1: Stakeholder prioritization

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 are thereby given a high priority.

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

Developers are given a medium priority since they have knowledge about whether a functionality is reasonable or not from a technical perspective.

Payment service was given a low priority as the payment service will be used only to provide the application with a standardized pay-functionality which is easily replaced.

The existing application *Riktiga Vykort* is a competitor to the application and is considered mainly as a benchmark and is hence given a low priority.

4.2.1.3 Tasks

Work area: Vacation
Communicating with friends and family. Usually from a remote location with unreliable internet access. Typically outdoors and overall poor working environment. Simplicity is key to capturing important moments.
Users: Average mobile 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 and recipients.
2. Search sent postcards and recipients.
3. Summary of cost.

Variants:

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

4.2.1.4 Interfaces

Req 1.2.2.1 Printer The interface connecting PostCardBuddy and the printed postcards is an off-the-shelf color printer able to handle standard image formats.

Req 1.2.2.2 Images Images for the standard gallery will be provided by the supplier of images in RAW-format.

Req 1.2.2.3 Permissions When users install the application they shall be prompted to grant permission to use the device functionallity specified in the context diagram.

4.2.2 Product

This section describes the functionality at the product level.

4.2.3 Images

Req 4.2.3.1 Image from phone gallery It shall be possible to choose pictures from the phone gallery for the front of the postcard.

Req 4.2.3.2 Picture from camera It shall be possible to take a picture with the camera and use as image for the front of the postcard.

Req 4.2.3.3 Image from standard library It shall be possible to choose pictures from the standard library.

Req 4.2.3.4 Image and GPS position Images from the standard library shall be presented based on the user's GPS position.

Req 4.2.3.5 Image editing The system shall have a function for editing images.

Req 4.2.3.6 Image saving It shall be possible to save images.

4.2.4 Greetings

Req 4.2.4.1 Greetings It shall be possible to write greetings in the app.

Req 4.2.4.2 Pictures of handwritten greetings It shall be possible to choose a picture of a hand-written greeting.

Req 4.2.4.3 Auto-generated greetings It shall be possible to choose a template greeting.

Req 4.2.4.4 GPS based greetings The system shall be able to generate greetings based on the GPS position.

Req 4.2.4.5 Handwritten greetings on screen It shall be possible to write a handwritten greeting directly on the screen.

Req 4.2.4.6 Saving greetings It shall be possible to save a greeting.

4.2.5 Recipients

Req 4.2.5.1 Enter recipients It shall be possible to enter recipients manually.

Req 4.2.5.2 Phone book recipients It shall be possible to choose recipients through the contacts of the device.

Req 4.2.5.3 Multiple recipients The system shall be able to handle multiple recipients for one postcard.

Req 4.2.5.4 Favourite recipients It shall be possible to save recipients as favourites.

Req 4.2.5.5 Frequent recipients The system shall show frequently used recipients as favourite recipients.

4.2.6 Postcard

Req 4.2.6.1 Saving postcards It shall be possible to save postcards.

Req 4.2.6.2 Reuse postcards It shall be possible to reuse saved postcards.

Req 4.2.6.3 Preview postcards It shall be possible to preview postcards before sending it.

Req 4.2.6.4 Digital postcard It shall be possible to send digital postcards.

Req 4.2.6.5 Physical postcards It shall be possible to send physical postcards.

Req 4.2.6.6 Payment It shall be possible to pay for sending physical postcards.

Req 4.2.6.7 Postcard size It shall be possible to choose the size of the physical postcard.

Req 4.2.6.8 Quality of physical postcard It shall be possible to choose the print quality of physical postcards.

Req 4.2.6.9 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.

4.2.7 Notifications

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.

4.2.8 Design

This section describes the design and layout of the application and postcards sent from it.

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

Req 4.2.8.7 Get image The application shall let the user choose the image source from a menu. See figure 3.

Req 4.2.8.8 Edit image The application shall give the user a basic image editor to customize the image. See figure 4.

Req 4.2.8.9 Recipiant address The application shall have an address input screen with an address-book/contacts (not in image). See figure 5.

4.2.9 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 7.

Figure 2: Start

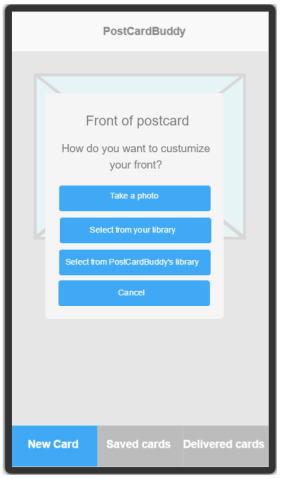

Figure 3: Chosee image source

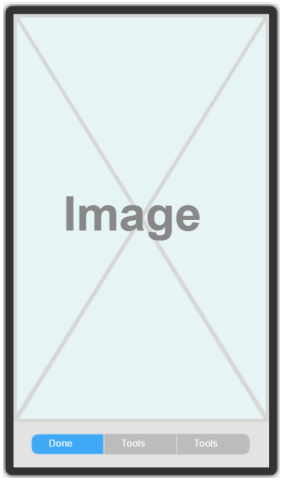

Figure 4: Image editor

Figure 5: Recipiant information

Figure 6: Those images are selected form what the key customer liked about the prototypes and should be used only as a guideline

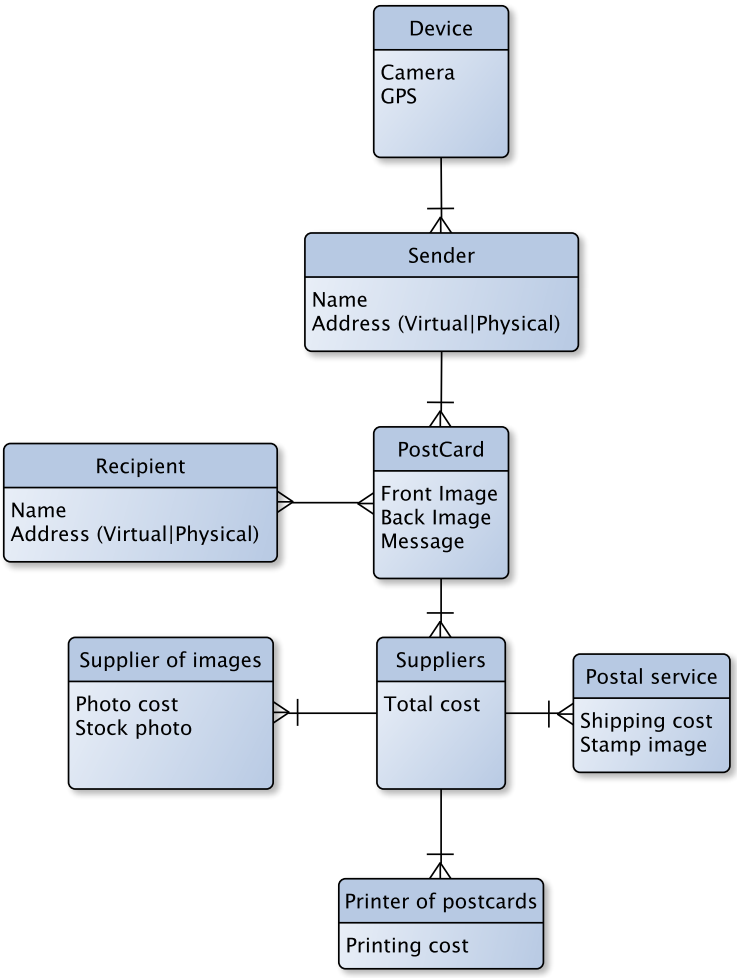

Figure 7: Data model

4.2.9.1 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:

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: Recipient

This class represents the person receiving the post card. This class is identical to *Sender* in terms of attribute structure. The sender and recipient 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 *Recipient*. 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 *Recipient* 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 *Postal service* to properly send the post card.

Class: Suppliers

This class collects the data from an *Supplier of images*, a *Postal service* and a *Printer of postcards*.

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 *Supplier of images/Photo cost*, a *Postal service/Shipping cost* and a *Printer of postcards/Printing cost*. The value is rounded up to two decimal places.

Class: Supplier of images:

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. A supplier of images with a photo and a cost.
2. A supplier of images 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: Postal service:

This class represents a postal service. The postal service 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 of postcards:
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.

4.2.9.2 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 8.

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 9.

Req 1.5.1.4 Recipient The input data to the *Recipient* class described in the Data dictionary shall include the items specified in the virtual window in figure 10.

4.3 Quality Requirements

4.3.1 Quality grid

This quality grid highlights quality factors for certain identified elements. Elements not in the grid can be seen as having status "As usual". Numbers in the quality grid are references to the numbers below the grid. The "x" does not have a reference. reference table 2

1. PostCardBuddy has a integrated payment solution for sending physical postcards. Integrity/Security shall be as usual for an app that contains an integrated payment solution.
2. Many users may use PostCardBuddy when they are traveling. Therefore it is important the app will work in a wide geographical area while still being reliable. Users sending digital, or physical, postcards not coming through will not be pleased.

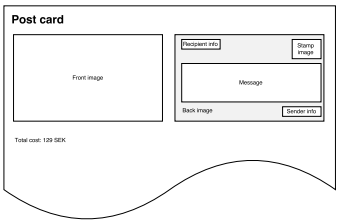

Figure 8: Virtual window PostCard

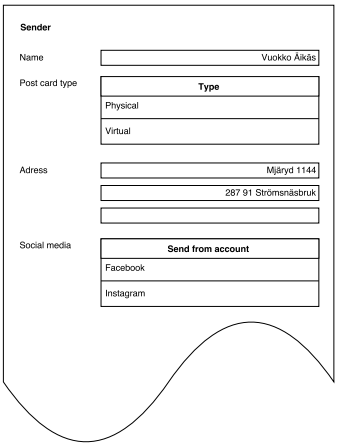

Figure 9: Virtual window Sender

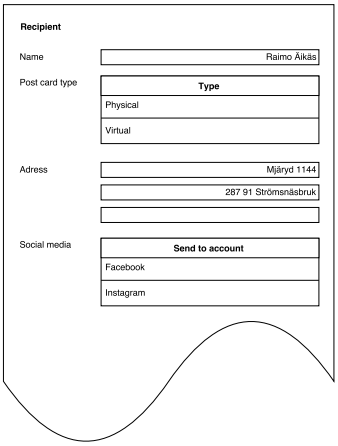

Figure 10: Virtual window Recipient

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			

3. The standard user might just send a few postcards a year. Because of this it is critical that the app is easy to use. Users should not need to go through unnecessary menu. It should be quick and easy to create a postcard and press ”send”.

4. Internet connection when traveling is often a problem. It is therefore critical that PostCardBuddy has a low internet demand. Many users may just use PostCardBuddy while traveling.

5. Users may install PostCardBuddy while traveling. Internet connection might be weak and users will probably want to do sightseeing, requiring them to leave internet zones.

6. PostCardBuddy saves files to the smartphone and uses features as the smartphones camera, and gallery. This should be smooth.

4.3.2 QUPER

A quper diagram takes into consideration quality and performace. This specific quper diagram measures time to get access to a specific image, from the image library, reference figure 11.

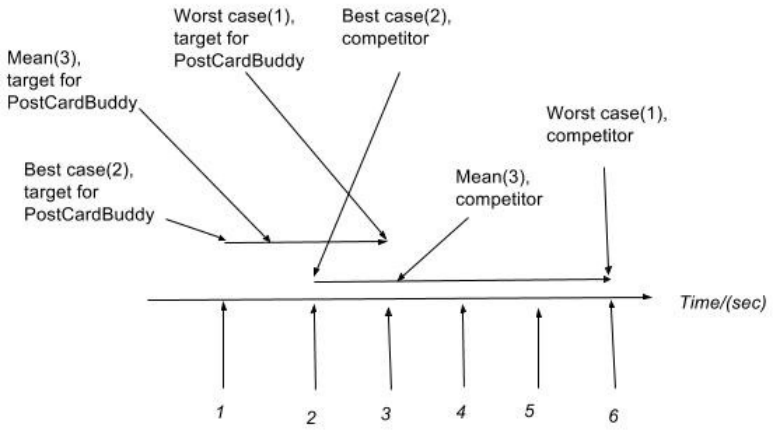

Figure 11: Time measured from requesting access to a specific image in the image library to it being loaded in to creating a postcard, with an internet speed of 30Mbps.
(1) For larger sized images. (2) For smaller sized images. (3) The mean of how long it takes.

4.3.3 Performance

Req 4.3.3.1 Memory usage The application shall adjust its memory usage depending on the device.

Req 4.3.3.2 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 4.3.3.3 Picture quality The camera shall be able to take a picture in the highest hardware supported resolution.

Req 4.3.3.4 Autofocus The camera shall have a autofocus that is comparable to the Android / iOS stock camera.

4.3.4 Availability

4.3.5 Security

Req 4.3.5.1 Store cards The photos shall be stored encrypted

4.3.6 Maintainability/Portability

Req 4.3.6.1 Language The application shall be developed in non native language e.g. Java for Android.

Req 4.3.6.2 Device support The application shall work on devices with newer operating systems than Android 4.1 / iOS 7.0.1

4.3.7 Usability

Req 4.3.7.1 User friendly 9 out of 10 users shall be able to use the system after a five minute instruction.

5 Release Plan