December 10, 2015 v2.04

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. The authors 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 efficient way to send personalized postcards directly from a smart phone. Simply choose an image for the front of the postcard, add a greeting on the back and send it to your friends and family. The image for the front can be a photo to capture a moment, an image from the phone's gallery or a template image. The template images can be found in the applications standard gallery. 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

Application The part of the product that is downloaded to the device.

Device Mobile device (e.g. smart phone) on which it is possible to download and use applications.

E-card An image and a greeting sent digitally presented to look like a physical postcard.

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.

Physical postcard A printed piece of paper with an image on the front, a greeting, recipients and postage at the back.

Postal service A company that delivers mail to private citizens.

Postcard The representation of physical postcards or E-cards within the application.

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 whom a postcard is addressed to.

Standard gallery Gallery of pre-existing images in the application.

Supplier of images The companies or people who supply the application with images for the standard gallery.

Template Greeting Pre-made suggestion for a greeting provided by the application.

Template Image Images in the standard gallery.

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:

1. Simplify the process of sending postcards
2. Enable user to send personalized postcards
3. Enable revenue generation for the printer of postcards

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 environment of the product and the stakeholders interacting with it. The demarcation between the inner and outer domain is based on whether the interaction with PostcardBuddy is direct. For example the printer is in the inner domain because the developer will have to take this interface into account when constructing the product.

There is one stakeholders that will interact directly with the application, namely the mobile user. 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 application's standard gallery. These images will be provided through an image database. The specification of the image database is out of the scope of this document. Images are provided to the database by a *Supplier of images*.

Some of the features of the product will require the use of functionality already 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 E-cards.

When a user sends a physical postcard it will be sent to a printer. The *Printer of postcards* has access to the printer and delivers the postcard to the postal service. Finally the postal service delivers the physical post card to the final recipients. *Printer of postcards* is responsible for a franking agreement with the postal service.

When sending physical postcards the user will be charged through a standard payment solution integrated in the product. The charge includes both the cost of the physical postcard and the franking.

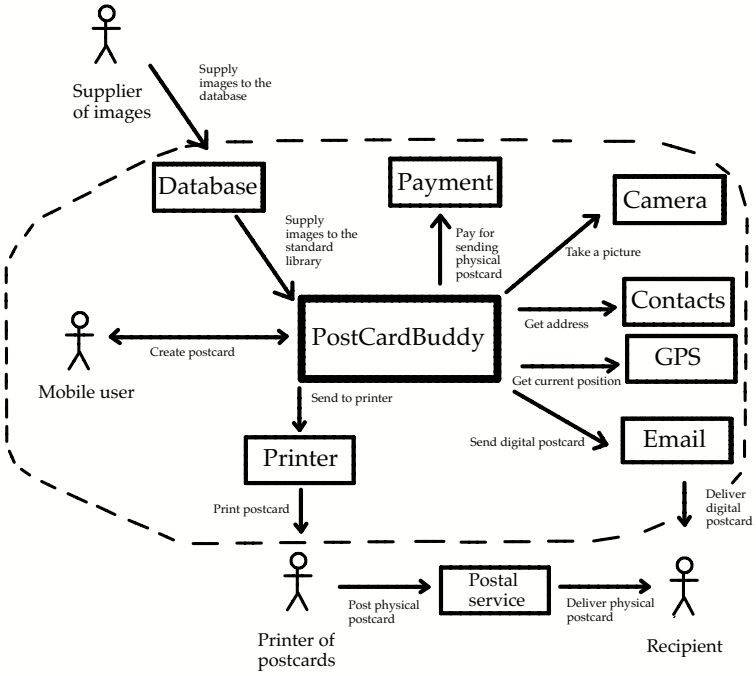


Figure 1: Context diagram of product.

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 actual 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 and is replaceable.

The existing application *Riktiga Vykort* is a competitor to the application and has only been used for comparison and is hence given a low priority.

4.2.1.3 Tasks

Req 4.2.1.1 The product shall support the following user tasks.

Work area: 1. 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 smart phone user, used to little manual work.

Work area: 2. Holidays
Communicating with friends and family. Usually in connection to holidays, e.g. Christmas. Personalization and simplicity is important.
Users: Average smart phone user, used to little manual work.

Task 1.1 Send a physical postcard.

Precondition: PostcardBuddy is running.

Sub-tasks:

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

Variants:

1a The user selects an image from the personal gallery.
1b The user selects an image from PostcardBuddy's standard gallery.
2a Includes editing of gallery images.
4a Choose a template greeting.
5a Manually add address.

Task 1.2 Send an E-card.

Precondition: PostcardBuddy is running.

Sub-tasks:

1. Use the camera in the device to take a picture.
2. Allow editing of images.
3. Save images.
4. Add greeting.
5. Save finished postcard.
6. Preview postcard.
7. Send postcard via E-mail.

Variants:

1a The user selects an image from the personal gallery.
1b The user selects an image from PostcardBuddy's standard gallery.
2a Includes editing of gallery images.
4a Choose a template greeting.
7a Publish postcard on social media.

Task 1.3 Send a physical postcard from a saved postcard.

Precondition: PostcardBuddy is running.

Sub-tasks:

1. Load a saved postcard.
2. Add missing parts to postcard.
3. Allow editing of postcard.
4. Add recipients from address book.
5. Preview postcard.
6. Send postcard to printer.

Variants:

4a Manually add address.

Task 1.4 History

Precondition: PostcardBuddy is running.

Sub-tasks:

1. View sent postcards and recipients.
2. Search sent postcards and recipients.

Variants:

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

4.2.1.4 Interfaces

Requirements related to the interfaces of the product.

Req 4.2.1.1 Printer The interface connecting PostCardBuddy and the printed postcards is an off-the-shelf color printer able to handle standard image formats.

Req 4.2.1.2 Images Images for the standard gallery will be provided by the supplier of images in JPEG, bnp or png format.

Req 4.2.1.3 Data of images The supplier of images shall tag the image with its GPS coordinates and the system shall be able to handle this data.

Req 4.2.1.4 Permissions When users install the application they shall be prompted to grant permission to use the device functionality specified in the context diagram.

Example: Camera, GPS etc.

4.2.2 Product

This section describes the functionality at the product level.

4.2.2.1 Images

Requirements related to the handling of images and all of them contribute to fulfil goal 1. Additionally, requirements 4.2.2.1, 4.2.2.2 and 4.2.2.5 contribute to meet goal 2.

Req 4.2.2.1 Image from phone gallery It shall be possible to choose pictures from the phone gallery for the front of the postcard.

Req 4.2.2.2 Image from camera It shall be possible to take a picture with the camera within the application and use as image for the front of the postcard.

Req 4.2.2.3 Image from standard gallery It shall be possible to choose pictures from the standard gallery.

Req 4.2.2.4 Image and GPS position Images from the standard gallery shall be sorted based on the user's GPS position.

Req 4.2.2.5 Image editing The product shall have the following functions for editing images; brightness, contrast, cropping, remove red eyes, filters, pen and text.

Req 4.2.2.6 Image saving It shall be possible to save images to the phone gallery.

4.2.2.2 Greetings

Requirements related to greetings, all of them contribute to fulfil goal 1. Additionally, requirement 4.2.2.7 contributes to meet goal 2.

Req 4.2.2.7 Greetings It shall be possible to write greetings within the application.

Req 4.2.2.8 Auto-generated greetings It shall be possible to choose a template greeting.

Req 4.2.2.9 GPS based greetings The product shall be able to generate greetings based on the GPS position.

Req 4.2.2.10 Saving greetings It shall be possible to save a greeting.

Req 4.2.2.11 Load greetings It shall be possible to load a saved greeting.

4.2.2.3 Recipients

Requirements related to handling of recipients.

Req 4.2.2.12 Enter recipients It shall be possible to enter recipients manually.

Req 4.2.2.13 Phone book recipients It shall be possible to choose recipients through the contacts of the device.

Req 4.2.2.14 Multiple recipients The product shall be able to handle multiple recipients for one postcard.

Req 4.2.2.15 Favourite recipients It shall be possible to save recipients as favourites.

Req 4.2.2.16 Frequent recipients The product shall show frequently used recipients as favourite recipients.

4.2.2.4 Postcard

Requirements related to functionality and handling of the postcard.

Req 4.2.2.17 Saving postcards It shall be possible to save postcards.

Req 4.2.2.18 Reuse postcards It shall be possible to reuse saved postcards.

Req 4.2.2.19 Preview postcards It shall be possible to preview postcards before sending it.

Req 4.2.2.20 Digital postcard It shall be possible to send digital postcards.

Req 4.2.2.21 Physical postcards It shall be possible to send physical postcards.

Req 4.2.2.22 Payment It shall be possible to pay for sending physical postcards.

Req 4.2.2.23 Postcard size It shall be possible to choose the size of the physical postcard.

Req 4.2.2.24 Quality of physical postcard It shall be possible to choose the print quality of physical postcards.

Req 4.2.2.25 History It shall be possible to display the history of sent postcards.

Req 4.2.2.26 Social media Feature for sharing postcards on social media.

4.2.2.5 Notifications

Requirements on how to manage notifications.

Req 4.2.2.27 Success notification The user shall be notified when an order is sent from a device.

Req 4.2.2.28 Fail notification The user shall be notified when an order fails to be sent from a device.

Req 4.2.2.29 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.

4.2.3 Design

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

The images in figure 2 are from a prototype and should be used as a guideline rather than as a design requirement.

Req 4.2.3.1 Front page The front of the postcard shall be a field containing an image.

Req 4.2.3.2 Text field The back of the postcard shall contain a text field.

Req 4.2.3.3 Address field The back of the postcard shall contain an address field.

Req 4.2.3.4 Postage field The back of the postcard shall contain a postage field.

Req 4.2.3.5 Postage print The postage shall be printed in the top right corner on the back of the postcard.

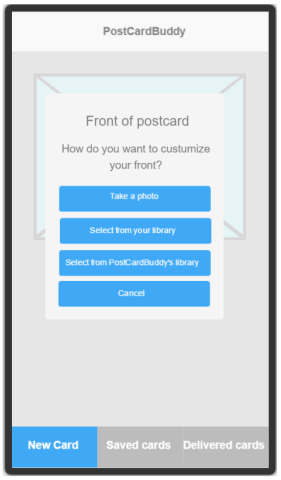
Req 4.2.3.6 Start Screen The application shall start with a screen where it is possible to choose front and back image/text, see figure 2a.

Req 4.2.3.7 Get image The application shall let the user choose the image source from a menu, see figure 2b.

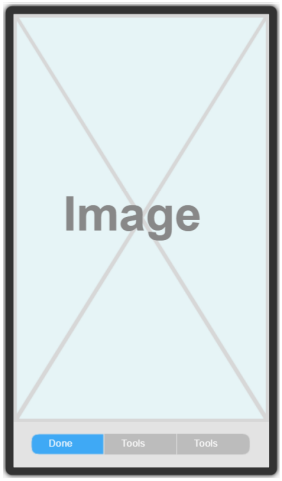
Req 4.2.3.8 Edit image The application shall give the user a basic image editor to customize the image, see figure 2c.

Req 4.2.3.9 Recipient address The application shall have an address input screen with an address-book/contacts (not in image), see figure 2d.

(a) Start



(b) Choose image source



(c) Image editor

(d) Recipient information

Figure 2: Example prototype of product.

4.2.4 Data Requirements

This section describes the data requirements of the application.

Req 4.2.4.1 Data model The product shall handle the data presented in the data model in figure 3.

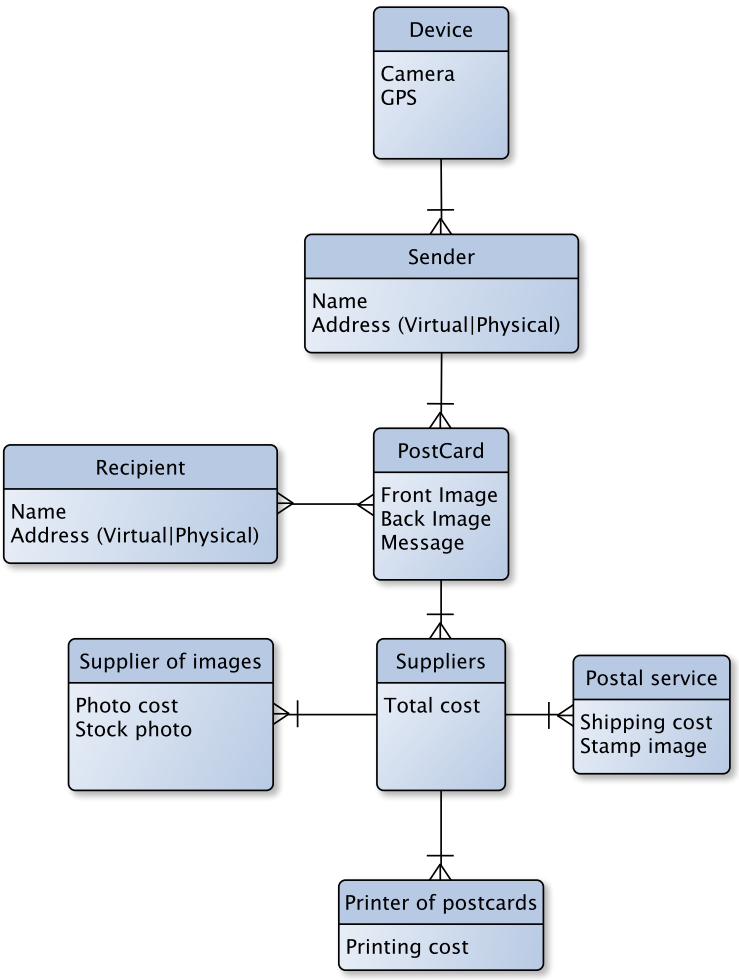


Figure 3: Data model

4.2.4.1 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 postcard. It can be the same person as the one using the device but it does not have to be.

Examples:

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

Attributes:

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

Class: Recipient

This class represents the person receiving the postcard. 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 postcard.
2. The same person as the one sending a postcard.

Class: PostCard

This class represents the postcard sent from the *Sender* to *Recipient*. It encapsulates all the information necessary to send a postcard 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 postcard.

Examples:

1. A postcard with two images, a message and a stamp.
2. A postcard 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 postcard.

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 postcard with Posten.
2. A representation of what is required to send a postcard 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 postcard to indicate that shipping was payed for.

Class: Printer of postcards:
This class represents a printer. The printer is responsible for printing the physical postcard.

Examples:

1. A company contracted to print postcards.
2. The company supplying the application.

Attributes:

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

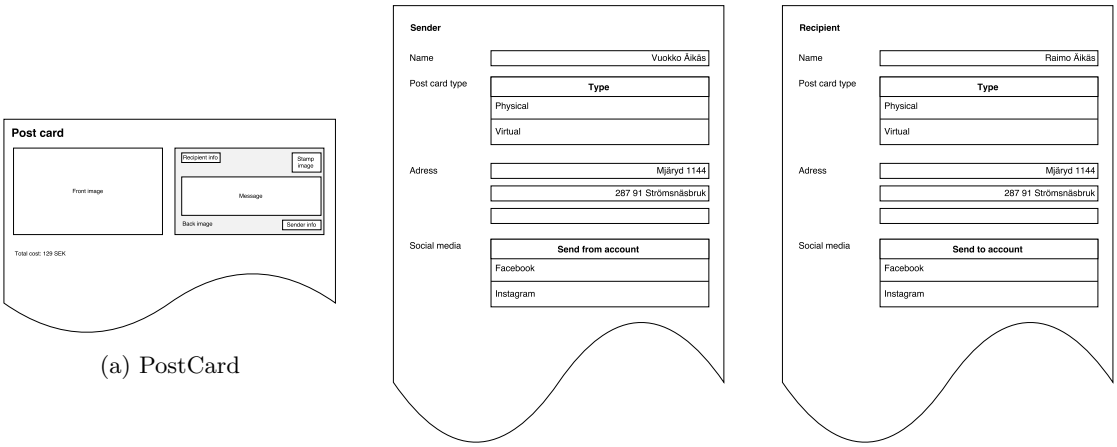
4.2.4.2 Virtual windows

Virtual windows displaying data of product can be seen in figure 4.

Req 4.2.4.2 PostCard The input data to the *PostCard* class described in the data dictionary shall include the items specified in the virtual window in figure 4a.

Req 4.2.4.3 Sender The input data to the *Sender* class described in the data dictionary shall include the items specified in the virtual window in figure 4b.

Req 4.2.4.4 Recipient The input data to the *Recipient* class described in the data dictionary shall include the items specified in the virtual window in figure 4c.



(a) PostCard

(b) Sender

(c) Recipient

Figure 4: Virtual windows displaying data of product.

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				
Miscellaneous					
Installability		5			
Interoperability		6			

4.3 Quality Requirements

4.3.1 Quality grid

This quality grid seen in table 2 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.

1. PostCardBuddy uses an integrated payment solution for sending physical postcards. Integrity/Security shall be as usual for an application that contains an integrated payment solution.

2. Many users may use PostCardBuddy when they are on vacation. Therefore it is important the application will work in a wide geographical area while still being reliable. Users sending digital, or physical, postcards not coming through will not be pleased.

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

4. Internet connection when on vacation is often a problem. It is therefore critical that PostCardBuddy has a low internet demand. Some users might only use PostCardBuddy on their vacation.

5. Users may install PostCardBuddy while on their vacation. Internet connection might be weak and disconnects from internet could be common. Also users might need to leave internet zones for trips on their vacation.

6. PostCardBuddy saves files to the smartphone and uses features as the phone's camera and gallery.

4.3.2 QUPER

A quper diagram takes into consideration quality and performance. This specific quper diagram measures time to get access to a specific image, from standard gallery, reference figure 5.

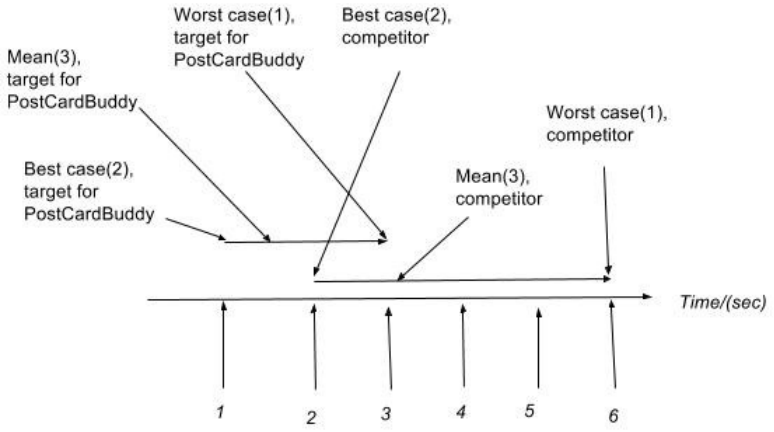


Figure 5: Time measured from requesting access to a specific image in the standard gallery to it being loaded in to creating a postcard. For the competitor this was done ten times with different pictures. Competitor, "Riktiga Vykort", was measured with a OnePlus One smartphone. The targeted area for PostCardBuddy should be done with the same kind of smartphone and tested in as a similar way as possible.
(1) Longest time measured. (2) Shortest time measured (3) The mean of the test.

4.3.3 Performance

All performance requirements shall be fulfilled on a Nexus 5/iPhone 5 running original firmware without any other user installed applications on the phone. Background tasks should be kept at a minimal by closing applications before testing.

Req 4.3.3.1 Application size The installed size of the application excluding user data shall not exceed 30MB.

Req 4.3.3.2 Speed The user interface shall respond within 200 ms after a finished user interaction if there should be an response. The test shall be done so that the above values can be guaranteed within a 95% confidence interval.

Req 4.3.3.3 Memory The application shall use less than 200MB RAM when editing a 8MP image. This is tested by using the all tools after each other which is one iteration. It passes the test if it can do 50 iterations without finishing the edit and uses less than 200MB RAM in the process.

Req 4.3.3.4 Picture quality The camera shall be able to take a picture in the highest hardware supported resolution. This requirement also applies to the latest iPhone/Nexus at the time of release for the application e.g. iPhone 6S (plus) and Nexus 6P/5X at 10/12/15.

Req 4.3.3.5 Autofocus The camera shall have an autofocus that is comparable to the Android/iOS stock camera. This means that the time to focus can't take longer than 0.5 sec more than the stock camera and the focus plane distance must be within 20% of the stock camera if the stock distance is less than 10m. The motive should be well lit and have good contrast so that the stock camera can find the same focus five times. The test shall be done so that the above values can be guaranteed within a 95% confidence interval.

4.3.4 Maintainability/Portability

Maintainability and portability of product.

Req 4.3.4.1 Language The application shall be developed in non-native language e.g. Java for Android.

Req 4.3.4.2 Device support The application shall work on devices with operating systems Android 4.1/iOS 7.0.1 or newer.

4.3.5 Usability

Usability of product.

Req 4.3.5.1 User friendly 9 out of 10 users shall be able to use the application after a five minute instruction.

5 Release Plan

Release	**Functionality**	**cost [h]**
1	Greeting	6
1	Physical postcard	120
1	Image from phone gallery	15
1	Payment	150
1	Picture from camera	24
1	Enter recipients	15
2	Image and GPS position	15
2	Preview postcards	30
2	Multiple recipients	15
2	Image from standard library	90
2	Phone book recipients	15
2	Auto-generated greetings	15
2	Handwritten greetings on screen	15
2	Pictures of handwritten greetings	30
3	Favorite recipients	9
3	Frequent recipients	9
3	Quality of physical postcard	9
3	Postcard size	24
3	History	15
3	Image diting	150
3	Saving postcards	15
3	Reuse postcards	30
3	Image saving	9
3	GPS based greetings	15
3	Saving greetings	9
3	Digital postcard	15
3	Social media	15

Table 3: Release plan

Release	**total sum [h]**
1	330
2	225
3	324

Table 4: Summary of the cost for each release