December 18, 2015 v3.00

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 to 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 application's 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.

Greeting The message the sender writes to the recipient on the back of the postcard.

Mobile user Person who uses 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 within 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 stakeholder 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 however the developers has to take its interface into account. 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 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.

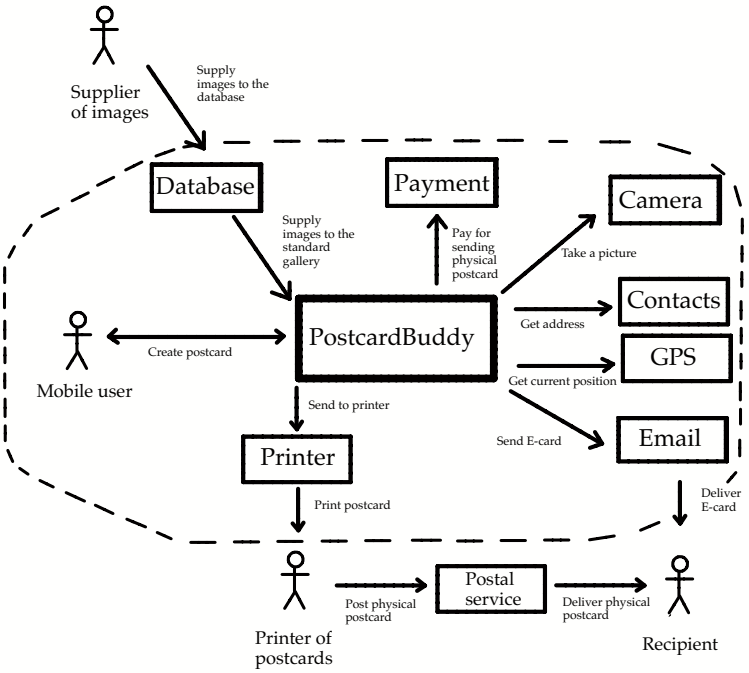


Figure 1: Context diagram of the product.

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.

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.

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

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. This requirement supports all three goal requirements.

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. Pay for postcard.
9. 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. Pay for postcard.
7. 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, contribute to meeting goal 1.

Req 4.2.1.2 Printer The interface connecting PostcardBuddy and the printed postcards is an off-the-shelf color printer able to handle standard image formats.

Req 4.2.1.3 Images Images for the standard gallery will be provided by the supplier of images in JPEG, BMP or PNG format.

Req 4.2.1.4 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 4.2.1.5 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.

Req 4.2.1.6 Postage The product shall be able to process an .xml file containing the postage information, provided by printer of postcards.

Example: Postage information includes cost of postage and image of postage that shall be printed on the postcard.

Req 4.2.1.7 Payment The product shall use an external payment service to charge for the services.

Clarification: The interface may vary depending on the payment service usedx.

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, contribute to meeting goal 1.

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.

Clarification: Favourite recipients shall be easier to access, e.g. by being placed at the top of the contact list.

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 postcards, all support goal 1.

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

Req 4.2.2.18 Load postcards It shall be possible to load saved postcards.

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

Req 4.2.2.20 E-card It shall be possible to send an E-card.

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

Req 4.2.2.22 Postcard size It shall be possible to choose between three different sizes of the postcard; small (A6), medium (A5) and large (A4).

Req 4.2.2.23 Social media Feature for sharing postcards on Facebook, Twitter and Instagram.

4.2.2.5 History

Requirements about displaying history, supports goal 1.

Req 4.2.2.24 View history It shall be possible to display sent postcards.

Req 4.2.2.25 View recipients It shall be possible to display the recipients to which a postcard has been sent.

Req 4.2.2.26 View date It shall be possible to display when a postcard was sent.

4.2.2.6 Payment

Requirements related to payment, supports goal 3.

Req 4.2.2.27 Payment It shall be possible to pay for sending physical postcards.

Req 4.2.2.28 Price It shall be possible to view the price before sending the postcard.

Req 4.2.2.29 Currency It shall be possible to choose the currency for the price to be displayed in.

4.2.2.7 Notifications

Requirements on how to manage notifications, supports goal 1.

Req 4.2.2.30 Success notification The user shall be notified when an order is sent from a device.

Req 4.2.2.31 Fail notification The user shall be notified when an order fails to be sent from a device.

Req 4.2.2.32 No internet If the user places an order on a device that is not connected to the internet, the order shall be stored and the next time the device receives internet connection a notification shall ask the user if he/she wants to try send it again.

4.2.3 Design

This section describes the design and layout of the application and postcards sent from it. The requirements support goal 1.

The images in figure 2 are from a prototype and should be used as a guideline rather than as a design requirement. Additionally, the prototype does not cover the entire application.

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

Req 4.2.3.2 Greeting The back of the postcard shall contain a text field for the greeting.

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.

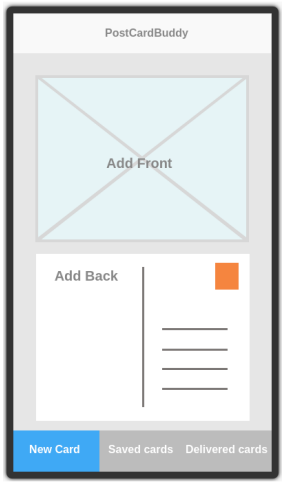
Req 4.2.3.5 Postage print The postage shall be printed in the top right corner on the back of the postcard.

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

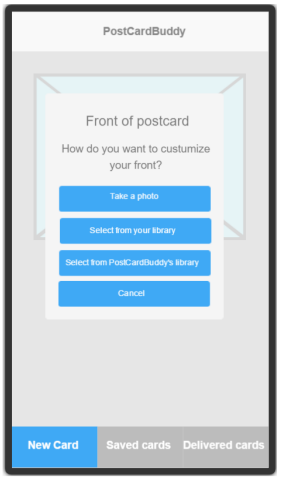
Req 4.2.3.7 Get image The application shall let the user choose the image source from a menu, see figure 2b.

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

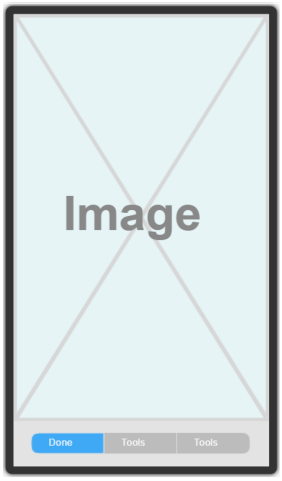
Req 4.2.3.9 Recipient address The application shall have an address input screen with an option of choosing contacts from a list (not in image), see figure 2d.



(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.

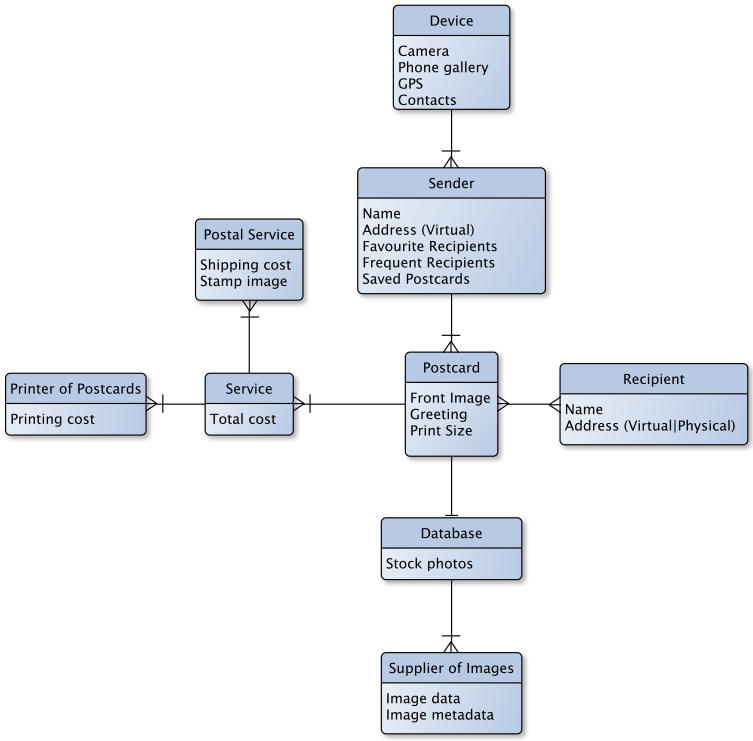


Figure 3: Data model

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 device's physical camera.
2. **Phone gallery:** List[Image]
   A list of images stored in the phone gallery.

3. **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.
4. **Contacts:** List[HashMap]
   The users local contact list. A contact is represented as a HashMap (Dictionary) of information.

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): String
   The sender's user name at a social media site or an email address.
3. **Favourite Recipients:** List[Recipient]
   A list of favourite recipients.
4. **Frequent Recipients:** List[Recipient]
   A list of recipients that the sender has previously sent postcards to.
5. **Saved Postcards:** List[Postcard]
   A list of the Senders saved postcards.

Class: Recipient

This class represents the person receiving the postcard. 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.

Attributes:

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

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 (E-card) or physical form. An instance of this object, owning a *Sender* and a *Recipient* needs to exist to be able to send a postcard. The image can come from the standard gallery or the phone gallery.

Examples:

1. A physical postcard with an image, a greeting and a stamp.
2. A physical postcard with no images, no greeting and a stamp.
3. An e-card with an image, a greeting and no stamp.

Attributes:

1. **Front image:** Image [optional]
   A compressed image that will be used as the front image of the *PostCard*.
2. **Greeting:** String [optional]
   The greeting on the *PostCard*.
3. **Print Size:** Enum [Small, Medium, Large]
   The size of a printed postcard. See Req 4.2.2.22 for definition.

Class: Service

This class collects the data from the *Postal Serice* and the *Printer of Postcards*.

Examples:

1. A collection of services relevant to printing and sending a specific *PostCard*.
2. Only a printing cost.
3. Only a postal cost.

Attributes:

1. **Total cost:** Float
   The combined cost of a *Postal service/Shipping cost* and a *Printer of Postcards/Printing cost*. The value is rounded up to two decimal places.

Class: Postal service:

This class represents a postal service. The postal service is the company that will transport the postcard.

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. **Stamp image:** Image
   The image supplied by *Postal service* to properly send the postcard.

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.

Class: Database:
A database used to store images supplied by *Supplier of images* in the standard gallery.

Attributes:

1. **Stock photos:** List[Image]
   A list of images (including metadata) supplied from *Supplier of Images*.

Class: Supplier of Images:
This class represents a supplier of stock photos.

Attributes:

1. **Image data:** Image
   The actual image that will be bought.
2. **Image metadata:** HashMap
   The metadata of the image, this includes the size and GPS tags.

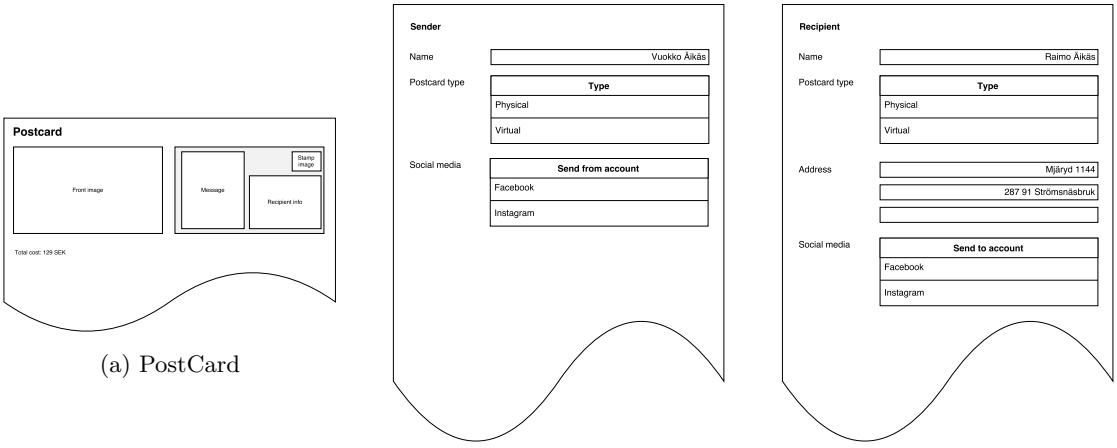
4.2.4.2 Virtual windows

Virtual windows displaying data of product can be seen in figure 4. The following requirements support goal 1.

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.

4.3 Quality Requirements

4.3.1 Quality grid

The 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.

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			

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 that the application will work in a wide geographical area while still being reliable. Users sending postcards that do not reach their destination 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.

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. In this Quper diagram we measured time from requesting access to a specific image in the standard gallery, to the image being loaded into creating a postcard. For the competitor, *Riktiga Vykort*, this was done twenty times with different pictures. The measurements were conducted on 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. For PostcardBuddy it would not be a disaster if it would do better than the Best Case. It is simply there to communicate that it does not need to do better as users would probably not care. It could also be seen as it is not worth it, in context of costs and trade-offs, to do better. See figure 5.

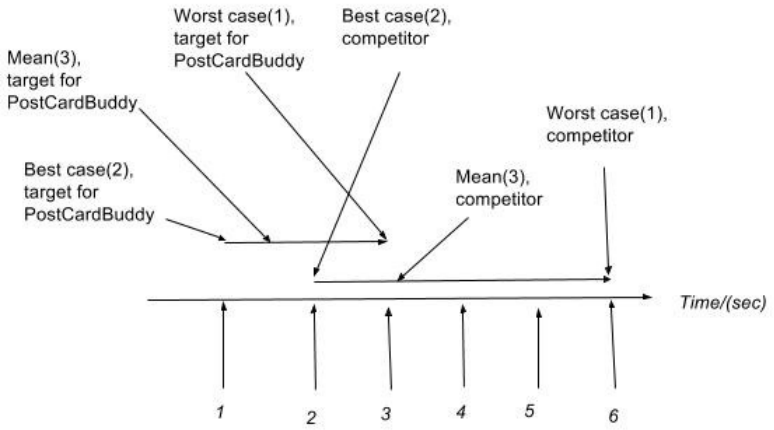


Figure 5: (1) Worst case, longest time measured. (2) Best case, shortest time measured.(3) Mean, the average time of the test.

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. The requirements support goal 1.

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

Req 4.3.3.2 User interface speed In 95 % of the cases, the user interface shall respond within 200 ms after a finished user interaction.

Req 4.3.3.3 Memory The application shall use less than 300 MB RAM when editing an 8 MP image with all tools used consecutively 50 times.

Req 4.3.3.4 Picture quality The camera shall be able to take a picture in the highest hardware supported resolution.

Clarification: This requirement 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. Within 95 % of the cases, the time to focus shall not take more than 0.5 s longer than the stock camera and the focus plane distance must be within 20 % of the stock camera if the distance is less than 10 m. The motive should be well lit and have good contrast so that the stock camera can find the same focus five times.

4.3.4 Maintainability/Portability

The following requirements are related to maintainability and portability of product and support goal 1.

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, supports goal 1.

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

A release plan has been developed taking into account the different stakeholders and their priorities. When designing the release plan the resources available for each release has also been taken into consideration. Furthermore, the cost of different functionalities has been approximated. The release plan of the product can be seen in table 3. Table 4 shows the cost in hours for each of the three releases.

Table 3: Release plan for functional requirements at product level.

Release	**Functionality**	**cost [h]**
1	4.2.2.7 Greetings	6
1	4.2.2.21 Physical postcards	120
1	4.2.2.1 Image from phone gallery	15
1	4.2.2.27 Payment	150
1	4.2.2.28 Price	9
1	4.2.2.29 Currency	9
1	4.2.2.2 Image from camera	24
1	4.2.2.12 Enter recipients	15
2	4.2.2.4 Image and GPS position	15
2	4.2.2.19 Preview postcards	30
2	4.2.2.14 Multiple recipients	15
2	4.2.2.3 Image from standard gallery	90
2	4.2.2.13 Phone book recipients	15
2	4.2.2.8 Auto-generated greetings	15
2	4.2.2.30 Success notification	15
2	4.2.2.31 Fail notification	15
2	4.2.2.32 No internet	15
3	4.2.2.15 Favourite recipients	9
3	4.2.2.16 Frequent recipients	9
3	4.2.2.22 Postcard size	24
3	4.2.2.**??** History	15
3	4.2.2.5 Image editing	150
3	4.2.2.17 Saving postcards	15
3	4.2.2.18 Load postcards	15
3	4.2.2.6 Image saving	9
3	4.2.2.9 GPS based greetings	15
3	4.2.2.10 Saving greetings	9
3	4.2.2.11 Load greetings	15
3	4.2.2.20 E-card	15
3	4.2.2.23 Social media	15

Table 4: Summary of the cost for each release.

Release	**total sum [h]**
1	348
2	225
3	315