

TEAM C

PostcardBuddy

Project Experiences

Authors of this document:

Emma Albertz
Caroline Brandberg
Linnéa Claesson
Billy Johansson
Johan Ju
Jacob Mejvik
Carl Rynegardh

Contents

1	Introduction	1
2	Methods and Techniques	1
2.1	Elicitation	1
2.2	Specification	2
2.3	Validation	3
2.4	Prioritization	3
3	Reflections	4
3.1	Elicitation	4
3.2	Specification	5
3.3	Validation	7
3.4	Prioritization	8
4	Personal Statements	9
4.1	Emma Albertz	9
4.2	Caroline Brandberg	10
4.3	Linnéa Claesson	10
4.4	Billy Johansson	10
4.5	Johan Ju	10
4.6	Jacob Mejvik	10
4.7	Carl Rynegardh	11
	Appendices	12
A	Prioritization using \$100 Method	12

1 Introduction

This document aims to describe how the work has been conducted during the project. It also contains the group's reflections on the work process and the difficulties regarding the different parts of the project.

2 Methods and Techniques

A description of the methods and techniques will be presented in this section, along with a short motivation for why the specific technique was chosen. In section 3 can evaluations of the used methods and techniques be found.

2.1 Elicitation

To find relevant elicitation techniques, *Software Requirements - Styles and Techniques* by Soren Lauesen has been used as a guidance[1].

The initial identification of relevant stakeholders emanated from a discussion about the product and who interacts with it. From this discussion the stakeholders that were the most important for this product could be identified. Since it was important to quickly get going with the project this method was considered appropriate. The group is well aware that this approach could cause important stakeholders to be left out. In order to reduce this risk, identification of other stakeholders has been a top priority throughout the project. The product is somewhat limited in scope and hence the number of additional stakeholders that have been considered significant have been few.

The following elicitation techniques were used:

Brainstorming Used as a first step within the team to come up with basic ideas and functions of the product. This is a quick method to get some initial ideas and starting points. During the brainstorming session the functions specified by the key customer, from their initial order of the product, were also considered.

Questionnaire The questionnaire was sent out to people within the end user group. Questions from the brainstorming session were used to form the questions. People answering were asked to grade functions from zero to five, where zero stood for not interesting and five for very interesting. An age field was added to see if there was a difference in interest of various functions between ages. This is also an easy method to get some ideas of what the intended users of the product might want (or not want).

Interviews In order to improve the understanding of the kind of product envisioned by the key customer, an interview session was conducted early in the elicitation process.

Efforts were made to contact the postal service to also conduct an interview with them. It proved to be too time consuming to get a hold of someone who could answer the questions and therefore this idea had to be dropped.

Prototypes Three team members created one prototype each, independently of each other so as not to affect each other's ideas. It was decided to do this right away due to the time constraint put upon this project. The prototypes are meant to be used for ideas to the graphical interface of the application. The use of prototypes is considered a suitable technique for this project since there are many easy to use and free programs available to create them. Additionally, it gives not only the stakeholders but also the authors of

the requirements a good idea of what it should look like and be able to do. They were specifically used when eliciting requirements from prospective end users.

Document study There is already a similar existing application on the market and it was used to further elicit functionality not already thought of and also to perhaps eliminate functionality that intervenes with the user experience. This was done *after* the initial brainstorming session, to avoid making an identical application or interfere with the creativity of the team.

2.2 Specification

Context diagram A context diagram was used since it is easy to make and is helpful when time comes for validation and verification. The diagram gives a good over-view of the system, both for the use of the client but also for the developers.

Data model A data model was one of the techniques used to describe the data requirements. It is a great way of quickly illustrating the overall structure of the system.

Data dictionary/Virtual windows The data dictionary was used to describe the data requirements in more detail. While the data model quickly gives an overview, the data dictionary is a bit more precise but does not illustrate relationships that well. The virtual windows is another great way of illustrating data input in specific scenes.

Task descriptions To clarify how the product is intended to work and what scenarios it should be able to handle, task descriptions were used. The task descriptions in the literature were not perfectly suited for our purposes and hence had to be tailored to the context.

Prototypes Pictures from the prototypes were used to give the developers a guideline on how the design could be for the application.

Quality grid A quality grid was created by analyzing which quality aspects were important for PostcardBuddy. This was done by starting out from Lauesen's quality grid[1]. Lauesen's quality grid was stripped down to the parts the group found worth mentioning. The following questions were used as guidelines for this:

- What quality aspects are more important for PostcardBuddy in comparison to other applications/software?
- Is there something that should be described more in detail to the developers?
- How are the quality factors ranked?
- How can the quality factors and their ranking be put into context? In other words, how can their importance be explained to the developers?

Several of the "As usual" quality factors were left out because the group thought that there was not much added value in elaborating further upon them.

QUPER A Quper diagram was created to specify a Quality requirement, or perhaps target is a better word. Quper is documented in ref [1]. Document studies of the competitor *Riktiga Vykort* showed that requesting images from the competitor's, *Riktiga Vykort*, image library into creating a postcard was a cumbersome process. If PostcardBuddy could be better in this aspect it would be highly valuable. Quper as in ref [1] contains advanced features requiring a lot of technical knowledge in the domain. Not having this knowledge, a lot parts

where left out so just the basics were left. To get a feel for what would be a reasonable target, in case of user experience, testing was done with *Riktiga Vykort*. As the group did not have enough knowledge for the technical aspect this was, more or less, left out.

2.3 Validation

Prototypes The prototype gives the customer a unique opportunity to validate how the product matches their expectations. The prototypes will be continuously adapted to the customer's needs and wants and new features will be added (or others removed) so that it becomes a good reflection on where the project is going.

Validation checklists and validation report (as developers) A validation checklist was constructed and given to the customer. They used the checklist to write a validation, to evaluate the System Requirement. This report was then used to go over the requirements and improve on them.

Validation checklists and validation report (as customers) Validation checklists provided by the developer group were used to validate the product initially ordered as customers. A validation report was then written based on the checklists.

Informal review An informal review within the group was conducted to make sure the System Requirements was complete, contained no ambiguity etc. Everyone read the report prior to the review. At the time of the review, needs for changes were discussed and a protocol was created listing everything that needed to be changed.

Validation by end user The requirements specification was given to potential end users to read through it. This was done to ensure that the requirements specification accurately describes the product.

2.4 Prioritization

Stakeholders The prioritization of the stakeholders was done through analysis and discussions within the group. The formal prioritization was done as one of the final tasks. However, the various stakeholders were given different amount of attention throughout the process as a natural consequence of the common view of the stakeholders within the group.

Release plan The release plan was done by first assigning a cost to every feature so that the cost would be as objective as possible. The list of features was then sorted according to the result from the \$100 method from the key costumer. The top five features were chosen for the first release to fulfill the key costumer's most important needs. Results from the end user questionnaire was then taken into account and added to the release plan accordingly. For every feature that was added the cost was taken into consideration and dependencies were analyzed, with help from the developers, so that they will be implemented in an order that is technically feasible.

Features To prioritize the features several techniques where used to cover different areas. End users were given a questionnaire where they could give a 1-5 answer based on how important they thought every feature was. To get hard data on what the key costumer wanted the \$100 method was used on two represents from the key costumer. Interviews where also held with the key costumer, to get a better understanding of the context and a clear overview of what was important.

3 Reflections

This section aims to evaluate the methods and techniques used, as described in section 2

3.1 Elicitation

Brainstorming The reason for selecting the method to collect our stakeholders was because it is a fast method which meant that it was possible to start working, such as contact some of the stakeholders.

Questionnaire Figure 1 presents the result of the questionnaire, which 38 people answered. To get answers from that amount of people was no problem and it gave a first idea of what the users were interested in. The result of this is that the functionality "Share postcard on social media" was not important and "Suggestion for GPS-based images" was appreciated. The result also showed that the desired functionality did not change that much depending on the age. Using a questionnaire was interesting since it gave a good idea of the functions people are interested in. However, as the questionnaire was created it was desirable that it was quick to answer. Therefore, only ten questions were used to maximize the number of respondents and the quality of the replies. Afterwards it was realized that some interesting functionalities were missing. Knowing the interest of these functionalities as well could have been of interest and might be investigated further prior future releases.

Interviews Although the interview provided valuable insights the main impression was ambiguity, both in terms of the role the key customer would have and exactly what the product should do. Given more time it would have been beneficial to invest in achieving a better consensus within the customer group before conducting the interview. Informal discussions were held with the key customer throughout the process.

Furthermore, two separate companies in the postal service business have been contacted with the intention to conduct interviews. However, it has proved difficult to get past the first line support and get a hold of an appropriate contact. A possible explanation for this is that the postcard business is only a minor part of the postal services market and there is probably nobody with a clear responsibility for this area.

Prototypes A program was used for constructing the prototypes that worked very well. It also proved to be of use for brainstorming new ideas and features, since the program itself offered a lot different options on how to do things.

From discussions with the costumer team, new ideas for features emerged when the costumer tried the prototypes. The prototype helped the costumer to verify that the application conformed to their requirements and also gave them an opportunity to see if something was missing or wrong.

Document study The existing application, and competitor, *Riktiga Vykort* was easy to use and rather slim. It did not contain a lot of functionality but it felt as there were enough. Most of the basic functions specified are already implemented in *Riktiga Vykort*. However, there are definitely some functionalities that could be of use that are not implemented. Also, *Riktiga Vykort*'s image gallery was not very big, and GPS based images depending on your localization only works in Sweden and Denmark. The group also noticed that it took a lot of time to request access to an image in *Riktiga Vykort*'s image gallery in to creating a postcard. The group started thinking that if PostcardBuddy would be able to do this better, it would be valuable giving competitive advantage. Therefore this was used in Quper.

Summary

In the elicitation process several different elicitation techniques were used. The group has continuously reflected on the choices and experiences from the various techniques. In some cases the reflections have highlighted the need for more information, resulting in that additional elicitation has been conducted. However, there are still some areas that require further elicitation as the development of the product commences. As an example, a specific area that deliberately has been kept somewhat limited is the exact technical specifications.

The elicitation process has also been heavily geared towards going beyond the initial stakeholders and challenge the domain borders. In doing so, several potential stakeholders, such as traveling companies and advertisement agencies, have been considered. The stakeholders that have been left out of the final project has been deemed to have little relevance to current structure of the project. In some cases, e.g traveling agencies, potential stakeholders could have been attributed with a higher relevance given a different business model for the product. Furthermore, in order to broaden and deepen the understanding of the domain stakeholders have been contacted. Although, this has given valuable feedback in most cases it also underlined some difficulties. For example, when contacting the postal services it proved difficult to get in touch with relevant staff to answer our rather technical questions.

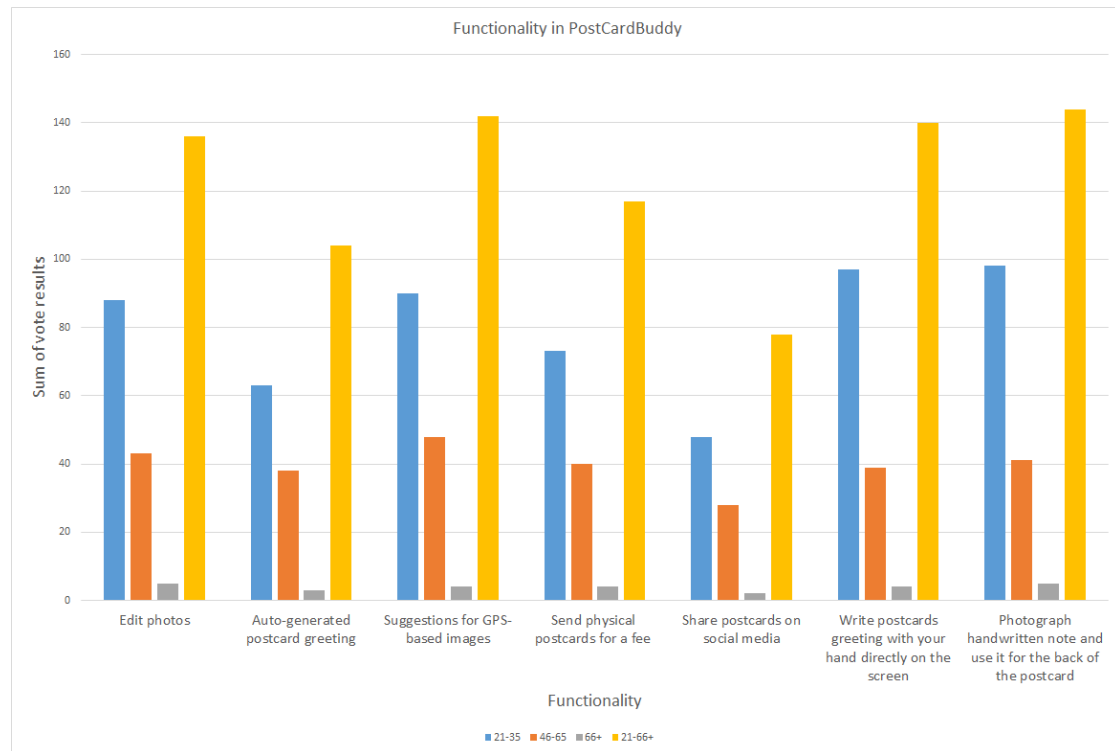


Figure 1: Result of the questionnaire on the desired functionality in PostcardBuddy

3.2 Specification

Context diagram The first context diagram created was presented in PMv2. The first diagram was very limited and contained too little information to understand the system. The updated diagram was then presented in release 1 of the report System Requirements. The biggest problem creating a context diagram was that it should be comprehensive enough to present important details, but small enough to be able to get an overview of the system. Therefore it is very important to think through which components it should contain, and which should be left out. This difference is often personal, which was noticed during the creation of release 1, which led to some discussions. Most of the discussions were spent talking about if the back-end should be presented and how the functionality that is used within the mobile should be presented.

The changes of the context diagram between release 1 and release 2 were mostly added descriptions of the parts. These descriptions were easily added without any problem. Also the contacts were added, which was a part that was missing in the previous version. It is very easy to miss parts of the diagram. To find out that every part is within, it is very good to try to describe each chain and see if it is easy from that description to follow in the context diagram.

Data model The data model is a good tool to easily visualize dependencies of different systems and stakeholders. If done thoroughly it could be used as a good starting point for developers, and in particular database developers. But the more complicated the data model becomes, the harder it gets for non-technical personnel to understand it. And in the same way it loses some of its value for developers if it is not thorough enough. This is why it was combined with a data dictionary and virtual windows, to adequately satisfy technical as well as non-technical personnel. Even if a high level of accuracy is too expensive, a low accuracy data model would help significantly with data requirements communication.

Data dictionary/Virtual windows The data dictionary is probably the simplest tool for specifications. It is easy to write but can become tedious and it is hard to see relationships between data. As a complement to a data model it is very good for properly communicating a specification. Virtual windows are very helpful for non-technical personnel and is a very efficient way of presenting an overview of what types of data are needed for a specific feature. It might be superfluous to use both virtual windows and a data model, but the combination of a data dictionary and one of the visualization tools would be recommended.

Task descriptions By giving a simple description in plain text the understanding of the product was increased through the use of task descriptions. This was especially useful when explaining to the user what the product should do. The drawback of the technique was that it required some adaptations from how it was described in the literature to fit our purposes. The main issue when tailoring the technique was to find an appropriate level of detail reflecting relevant information about the product and its use. The final design was largely impacted by the validation process and the feedback from users reading the specification.

Quality grid Doing a quality grid was a good idea. While creating the quality grid many ideas came to life, especially ideas for different requirements. The quality grid should probably have been created earlier in the project phase. Maybe even for release one. Discussing the quality grid more with the stakeholders, e.g. using questionnaires or interviews, could have been beneficial.

Creating the quality grid forces the creator to think of different aspects for the application. The creator gets an overview of what is important, and with that ideas can come to life. Therefore, a quality grid could be a way of elicitation, mostly quality requirements, but maybe even functional requirements. The line between quality requirements and functional requirements can be very thin. A negative aspect with the Quality grid could be that it does not specify any requirements. It mostly highlights what is important for the application and describes to the developer why this is important. If it really is a negative aspect could be discussed, as it is a very useful way of communicating quality factors to the developers. As the Quality grid was created rather late in the project, not many quality requirements were generated from it. If the quality grid would have been created earlier, more quality requirements could have been based on the grid.

QUPER It is interesting to compare the quality grid with Quper. The quality grid gives a better overview over quality factors as a whole while Quper serves as a way to specify a target or requirement. Normally, Quper demands a lot of knowledge which the group did not have. Therefore, it was not very useful in our case. It did highlight the problem with the competitor *Riktiga Vykort*'s application and specified how long time it should take for PostcardBuddy, but this could have been done in different ways. If we would have more knowledge in the domain, Quper would probably have been more useful.

Summary

When specifying the requirements numerous different techniques have been used. Furthermore, the requirements have been specified based on type and abstraction level used to describe the requirement. A very important aspect of this specification has been to construct a precise understanding of the context of the product. There has been many discussions and revisions of the systems boundaries before arriving at the final context diagram. In general the overall experience from the specification is that it is impossible to specify everything with a great level of detail. The solution to this problem has been to combine different degrees of completeness and different abstraction levels. It has also been helpful to use several different specification techniques, tailored to the situation at hand, combined with examples.

In addition, the group has paid special attention to tracing the requirements to the goals while simultaneously maintaining different hierarchies and requirement relations. When it comes to establishing a desired level for the target quality the QUPER analysis has been of great help, since it provides a good framework and scale to benchmark against competitors.

Perhaps the most challenging aspect of the process has been to prioritize and decide on which aspects need special attention. Since one of the overarching goals of the product is simplicity, special attention has been given to achieve this. In practice this means that it has been important to specify exactly what data needs to be added and special attention has been given to this aspect.

3.3 Validation

Prototypes The prototypes worked well and the key costumers appreciated when they could see how the application may look.

Validation checklists and validation report (as developers) It was very useful to receive input from the key customer. They noticed things previously missed and had very clear and concrete comments that were easily fixed.

Validation check lists and validation report (as customers) The use of checklists was very helpful as it is easy to check the quality of existing requirements. However, it is also lim-

iting in the sense that it makes it difficult to see if there are any missing requirements. It is easy to "get stuck inside the box" that the checklist actually is. Checklists are very quantitative and need clarifying comments in addition to get qualitative information from them.

Informal review This was very important in the process. Since everyone has been working on separate parts it was easy for contradictions to appear. Additionally, it gave a good overview of the state of the report such as gaps that needed to be filled etc. It would probably have been a good idea to do this more continuously throughout the project, to make sure everyone was headed in the same direction.

Validation by end user Letting potential end users validate the requirements specification was useful since it can be difficult to distinguish inconsistencies in one's own work. This definitely contributed in improving and clarifying the requirements specification. Their view of the product was consistent with the intentions, hence few alterations were necessary.

Summary

The process of validating the system requirements specification has been an ongoing process that has been intertwined with the elicitation process. During this process a range of problems associated with the quality of the system specification has been found. The group has focused a lot on trying to engage the stakeholders in the validation, through providing prototypes and showing the requirements specification as it evolved. To increase the quality further, informal reviews have been held within the group. It has been stressed that requirements should be both accurate and testable at an appropriate level.

When acting as a key customer it was important to give useful feedback that would go beyond form issues. Since the group developing our product provided a good checklist the validation was quite simple in that sense. The checklist did however cause some troubles as it gave a very strict frame for validating and might have resulted in that some problems were missed.

3.4 Prioritization

Stakeholders Some parts of the prioritization process of the stakeholders were relatively easy, whereas others were more difficult. The stakeholders identified as the most important are the ones that would in some way interact with the system. For example the mobile users were considered as the most important stakeholder as the product is entirely dependent on that they are willing to use it. A major difficulty was to handle the situation when one stakeholder with a low priority had a strong opinion about one aspect of the system that collided with a weaker opinion of a higher prioritized stakeholder. For example, the developers had strong opinions on what should happen when a user tries to send a postcard and the internet connection does not work.

A learning outcome is that prioritization should have been done at an earlier stage of the project, especially of the stakeholders. For this project though, it did not cause any major issues since the group members all had similar views despite the lack of a formal prioritization.

Release plan The first problem that occurred for the release plan was to establish the cost estimates. Different members had different cost estimates for the features as a result of previous experiences in mobile application development. Some had experience developing for the android platform while others had developed for iOS. It is reasonable that features

are more or less difficult to implement on one platform than another, but it was decided to use an average cost as an estimate. The dependency problem was easier to solve as there were not many dependencies to begin with and the dependencies that were there had a low cost and could be added easily. To handle constraints in the release plan it was decided that the cost of different releases should be quite evenly distributed while taking into consideration which features were essential, important or "good to have".

There is a tool to help the specification of requirements and release planning, called reqT, that was recommended to use in this project. After the cost and benefit were estimated, the data was put into reqT to try and solve the optimization problem. While working with reqT it was learned just how hard the optimization problem can be to solve. In fact reqT could not solve it within a reasonable time span without manually modifying the problem. Due to inexperience with the tool it was decided to abandon it and do the release planning manually. In this project this was reasonable since the amount of requirements was manageable.

Features The \$100 method was considered a good technique when prioritizing features, however some things could have been done differently. For example, letting the customer grade a basic feature essential to the product, against a feature that could improve the user experience is not a good technique. If the basic features would be graded separately from the external features, the results would be more useful.

In addition to the \$100 method, the replies to the questionnaire were also used. Although the users did not have the opportunity to grade all the final features, their input was considered important and was therefore taken into account.

One of the difficulties was constantly changing requirements, which meant that some of the prioritization that had been done was not entirely accurate anymore.

Summary

Although prioritization has been done in previous courses the formalized approach in this course was a new experience that turned out to be rather challenging. Several different techniques were used but despite that they provided a framework, there was still a need to take a lot of decisions within these techniques. For example when combining priorities from different stakeholders resulted in a need to go beyond the ordinal scale in the sense that a low prioritized stakeholder at times had very strong opinions of certain aspects. In spite of these challenges a release plan based on the prioritization was created. The release plan has been developed iteratively taking into account both scheduling and precedence constraints. The process of finding the right priorities has also been very helpful in directing attention to areas that needed special attention in order to improve the quality of the specification.

4 Personal Statements

4.1 Emma Albartz

In this project I was assigned the role as Elicitation and Prototype Manager (EPM). During the elicitation process I was involved in the work of collecting requirements from the key customers and the postal service. The intention was to do this by conducting interviews with both of these stakeholders. However the process of getting hold of a representative at the postal service that could answer our questions proved to be very time consuming. Since the time frame for the project was limited we were forced to abandon this idea after extensive email conversations. The

work during the elicitation process over all contributed to the functional requirements part in the requirement specification. Towards the end of the project I was also in charge of preparing and holding the conference presentation.

4.2 Caroline Brandberg

Acted as SCCVM (Stakeholder, Customer Communication and Validation Manager). Handled the communication with the group where we acted as key customer. Also responsible for the context diagram with related text as well as the stakeholder analysis. Both of these in correlation with valid input from team members. Create a \$100 method-form based on the functional requirements at product level, and sent the form to key customers. Contributed in producing the release plan in correlation with group members. Also created a prototype, contributed in creating the questionnaire, responsible for receiving 10 answers to the questionnaire, compiled the responses from the questionnaire in a diagram, participated in the discussion with key customer about the prototypes and participated in the informal review.

4.3 Linnéa Claesson

I was assigned the role of project manager. My main responsibilities were to keep track of what needed to be done and when. I continuously read through the documents throughout the process to find contents and quality inconsistencies. Considering we were seven people working separately a lot of the time, these inconsistencies were bound to happen. I tried to fix the simpler ones along the way, prior the releases mainly, and assign the rest to the group members responsible for those areas.

Other parts where I contributed were goal tracing, validation as customer using the provided checklist and writing a report, validation as developer going through the validation report written by the customer and using it as a basis to make changes to the System Requirements and general contributions to writing requirements.

4.4 Billy Johansson

As the DRM my main responsibility was the data requirements section. This included the production of the data model, data dictionary and virtual windows, as well as verifying that they were consistent with the rest of the requirements.

I also helped producing the questionnaire used for elicitation, collected answers, made a prototype application as well as helped with prioritization and the release plan. I also participated in group meetings and informal reviews.

4.5 Johan Ju

As QRM my main responsibility was the quality requirements but I also managed the prototypes. I also contributed to the formulation of some functional requirements, worked with the prioritization and release plan. Under the elicitation phase I helped with the questionnaire and had discussions with the key customer about the prototypes. I have also actively participated in the inspection and amendment of the specification.

4.6 Jacob Mejvik

I was assigned the role of tools, documents, experiences and version manager. However, the division of labor within our group has not followed the roles strictly and I have been involved

in a number of different areas. The tasks where I have directed most of my attention has been elicitation, task descriptions and structuring the final reports. I was also assigned the task of preparing and conducting the final presentation.

4.7 Carl Rynegardh

Contributed in/with: Questionnaire, functional requirements, quality grid, quper, validating another groups requirements, informal reviewing.

References

- [1] Soren Lauesen, *Software Requirements - Styles and Techniques*, Pearson Education Limited, 2002
- [2] Björn Regnell et al, *Supporting Roadmapping of Quality Requirements*, published in IEEE Software, 2008

Appendices

A Prioritization using \$100 Method

Images

Image from phone gallery - It shall be possible to choose pictures from the phone gallery for the front of the postcard.
Picture from camera - It shall be possible to take a picture through the camera and use as image for the front of the postcard.
Image from standard library - It shall be possible to choose pictures from a standard library.
Image and GPS position - Images from the standard library shall be presented based on the user's GPS position.
Image editing - The system shall have a function for editing of images.
Image saving - It shall be possible to save images.

Greetings

Greetings - It shall be possible to write greetings in the app.
Pictures of handwritten greetings - It shall be possible to choose a picture of a hand-written greeting.
Auto-generated greetings - It shall be possible to choose a template greeting.
GPS based greetings - The system shall be able to generate greetings based on GPS position.
Handwritten greetings on screen - It shall be possible to write a handwritten greetings directly on the screen.
Saving greetings - It shall be possible to save a greeting.

Recipients

Enter recipients - It shall be possible to enter recipients manually.
Phone book recipients - It shall be possible to choose recipients through the phone book.
Multiple recipients - The system shall be able to handle multiple recipients for one postcard.
Favourite recipients - It shall be possible to save recipients as favourites.
Frequent recipients - The system shall show frequently used recipients as favourite recipients.

Postcard

Saving postcards - It shall be possible to save postcards.
Reuse postcards - It shall be possible to reuse saved postcards.
Preview postcards - It shall be possible to preview postcards before sending it.
Digital postcard - It shall be possible to send digital postcards.
Physical postcards - It shall be possible to send physical postcards.
Payment - It shall be possible to pay for sending physical postcards.
Postcard size - It shall be possible to choose the size of the physical postcard.
Quality of physical postcard - It shall be possible to choose the print quality of physical postcards.
History - It shall be possible to display the history of sent postcards.
Social media - Feature for sharing postcards on social media.

Total sum (\$):

\$ - Stakeholder 1	\$ - Stakeholder 2
10	40
10	12
0	20
10	11
0	2
0	0
20	35
0	0
0	2
0	0
0	2
0	1
0	20
5	5
10	0
0	15
5	1
0	4
0	4
10	5
0	0
10	60
0	60
3	0
5	0
2	1
0	0
100	300