Project: Panna

General Requirements

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym	Panna
Project Full Name	The General Requirements of the Project Panna
Storage	Microsoft office Sharepoint 365
Format	PDF, docx
Dissemination	All Working Members
Start Date	09/April/2016
Iteration	1
Version	0.01
Responsible	Vitali Dettling
Email	vitali.dettling@gmail.com

Table of Content

1 DOCUMENT HISTORY

Version	Date	Changes	Author(s)
0.01	09/April/16	Initialization of the document: Requirements and Booklets	Vitali Dettling
0.01	09/April/16	Init: Project Management	Vitali Dettling
0.01	09/April/16	Init: Introduction	Vitali Dettling
0.01	13/April/16	Init: Functional Requirements	Vitali Dettling
0.01	21/April/16	Init: Non-Functional Requirements: Core Features and Foundation	Vitali Dettling
0.01	23/April/16	Non-Functional Requirements: Scenarios	Vitali Dettling
0.02	22/August/16	Updated the whole script, in particular the backend specifications (section 6)	Vitali Dettling

2 GLOSSARY

Term	Explanation
Panna	Name of the project, as well as an analogy to the project content.
B2B	Business to Business
B2C	Business to Customer
QoF	Quality of Service
SOA	Service Oriented Architecture
QR-Code	Quick Response Code
SSO	Single Sign-on. One way token generation to access multiple separated application within a single environment.
Authentication	Used to verify an identity (user, device, system) and find out, who is calling a service.
Authorization	Used to determine what an identity is allowed to do and check whether a caller is approved to call a service or see its results.
Confidentiality	Confidentiality: ensure that data remains confidential and that no one besides the caller can see service data during transmission
Integrity	Ensure that data cannot be manipulated or counterfeited and that neither service data is wrong nor credentials for authentication or authorization of a caller are manipulated.
Availability	Ensure that a fully functional service stays available and is not compromised e.g. by flooding it with requests in a denial of service (DoS) attack.
Accounting	Keep track of the consumption of resources and track service calls e.g. for management, planning or billing purposes.

Auditing	Evaluate security concepts and implementations to improve reliability. The goal is to detect or analyze security-holes and attacks. This includes monitoring, logging and tracing of security-relevant data flow.
WS-* security standards	● WS-Security ● XML Signature ● XML Encryption ● XML Key Management (XKMS) ● WS-SecureConversation ● WS-SecurityPolicy ● WS-Trust ● WS-Federation ● WS-Federation Active Requestor Profile ● WS-Federation Passive Requestor Profile ● Web Services Security Kerberos Binding ● Web Single Sign-On Interoperability Profile ● Web Single Sign-On Metadata Exchange Protocol ● Security Assertion Markup Language (SAML) ● XACML

3 INTRODUCTION

The main objection of this document is to provide an overview of the Panna project. The general features of the product is to provide a platform where customers can store, sort and share content, e.g. images. The product should be easy in use; in such a way that non technical-affine people are able to use it. Furthermore, the product provides functionalities to store the content encrypted as well as on different stores. Those storages could be their own servers or third party products, such as the cloud. Finally, the product has to be performant in speed and it should be usable on all platforms.

3.1 Document Structure

The document is following the german standards for “Lastenheft” which can be translated to “product concept catalogue”. Each chapter describes certains aspect of the product, whereas each chapter and/or sub-chapter can have their own product concept catalogues. There is one exception, the chapter with the current state of the project is not considered here because the product is going to be developed from the beginning.

Moreover, each document has a history of all contributors to the document. This is important for three reasons. Firstly, to see who has contributed, to which chapter and to what date, in order for future questions. Secondly, to have an overview when the last time the document has been changed. And thirdly, some information are most likely to change over time, due to the nature of things. Hence, the document should be adjusted accordingly each time.

Last but not least, the structure of the document should be followed due to the previous contributor. In order to avoid multiple different types of configuration setting, e.g. to avoid typographical diversity. Finally, the main goal of the document is to provide a clear understanding, as well as agreement, of the current and future state of the project for all project members. Hence, it should be written unambiguous and as short as required.

3.2 State-of-the-Art

There are other encryption and sharing programs, namely:

- WhatsApp: [1]
- OpenKeyChain: [2]
- Threema: [3]

These programs, can be used as a source for inspiration or references for working examples. The sources could be in form of papers or source code; if open sourced of course.

4 FUNCTIONAL REQUIREMENTS

The functional requirements specifies what the Panna project should do; in a more detailed but still abstract manner. Therefore, this chapter is divided into multiple parts to describe the project through different perspectives.

4.1 Features

These features should be included in the Product:

- Storage of data: this includes Panna server as well as third party servers
- Sorting and tagging of images through customer (Moments)
- Sharing of data with other registered people
- Encryption of data

4.2 Administration

The Panna product should be as easy to configure as possible, because of the B2C requirement. In fact, the customer should not know that he/she is configure the product at all. In order to do that the Panna project should have a great focus on user friendliness. This also means the product should be pre-configured, hence it should work out of the box.

4.3 Privacy

The product is not storing any types of data from the customer, except for the registration purpose and the location of the third party storage. These information are stored in the Panna DB securely.

The required information are, but not limited to:

- Authentication (Name, Surname, email, etc.)
- Address (City, Post code, etc.)
- Payment details

4.4 Devices

The product should work on different devices:

- Computers
- Laptops
- Mobile devices
- Tablets
- On all operation systems

4.5 Security

One of the main feature is the encryption of the stored data. In order to do that one need to manage with keys (private and public keys). Meaning, the customer has to store the keys at his/her local device to guarantee that nobody can see the data content. Despites the fact that the keys are stored locally the customer should not be aware of them. Hence, to avoid data lost, due to computer crashes, the customer should have multiple options to manage his/her keys:

- Printing the secret keys on a paper
- Digital copies and distributed at his/her devices through QR-codes
- Storing the secret keys on the Panna server

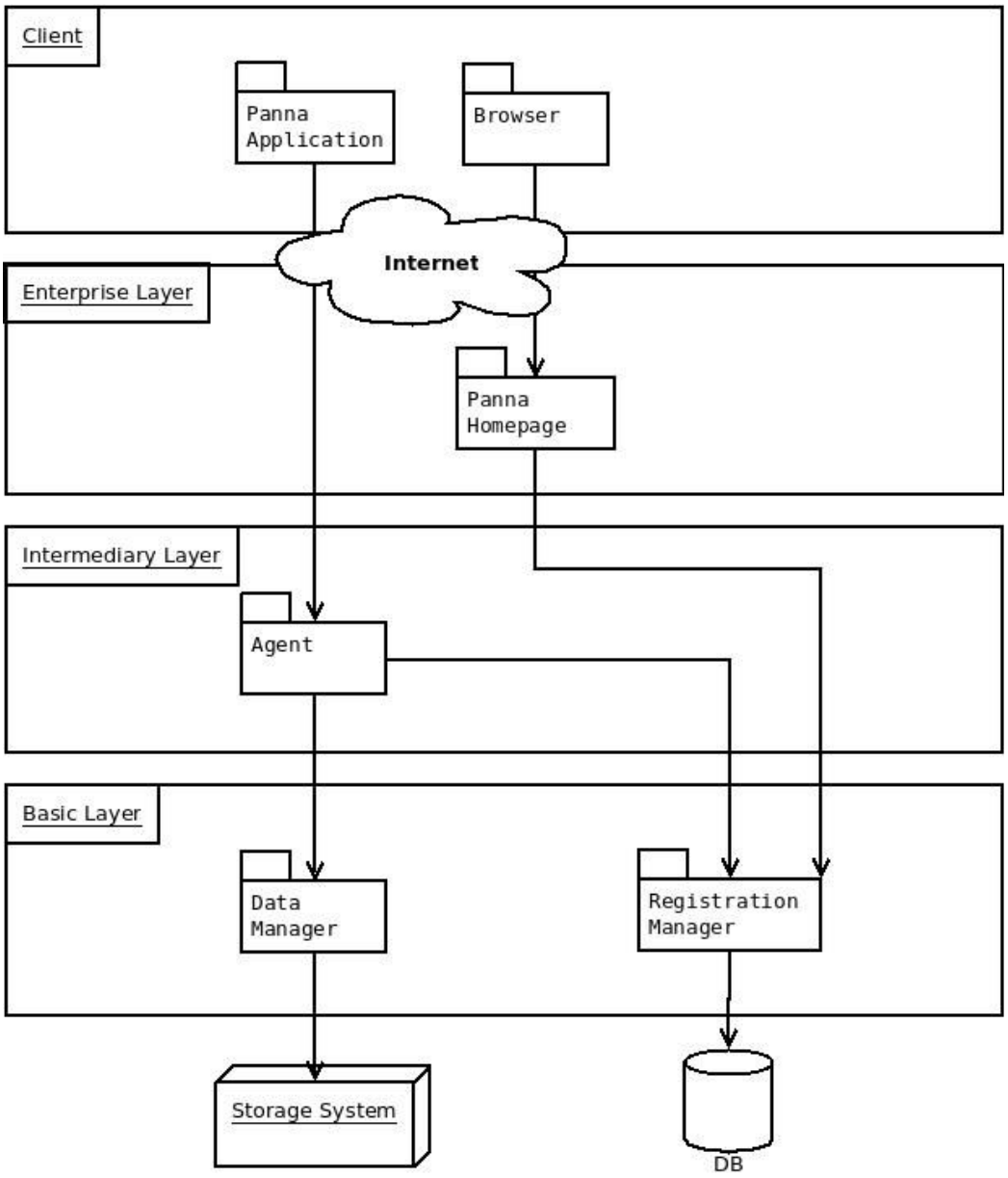
4.6 Homepage

Finally, a homepage are required to provide information for the customer. Moreover, it should be a reliable location where a potential customer can download the Panna software. For QoF (Quality of Service) especially during the alpha and beta releases, the homepage should have possibilities to communicate with our staff members or other customer. This can be done through blogs or chats. Anyway, the homepage needs to be developed before the alpha release. The concrete details will be specified in an another requirement document.

5 NON-FUNCTIONAL REQUIREMENTS

This chapter describes how the functional requirements will be implemented. In order to do that it is splitting the project into smaller junks. These junks will be analysed and splitted again if necessary. Furthermore, the description of the core functionalities is following the SOA (Service Oriented Architecture) naming convention. Finally, the model is taken from the “Federal Office for information Security” [4].

5.1 Core Features and Foundation



Client

- The client layer contains the application for the user to access the Panna project.
- Through the browser one can download the software.
- Panna Application en- and decrypts data. It also manage the keys within the user's environment. Moreover, it provides an Agent in order to simulate the servers behaviour by network lost. It also runs performance consuming algorithm to reduce the quality of images for instance, in order to save bandwidth. These have to be done without the user's knowledge and without to interrupt the user's other programs.

Enterprise Layer

- The Panna Website creates a platform, in order to download the software, get information and to give feedbacks.

Intermediary Layer

- The Agent is a design pattern to hide the connection to the client. Hence it simulates the connection in case the network is down. It also checks the incoming data via a checksum. According to the check an exception is thrown or not. The exact interface specification can be found in the next chapter. Keep in mind, it cannot compress the data for performance reasons, otherwise the checksum would be corrupted.

Basic Layers

- Management of stored data via a SQL DB
- Stores user credentials and cached them if necessary.
- Manage actual data in a storage system
- Synchronised with the servers deposited data
- Manage user registration via SSO and tokens
- Check access validity of an customer via a SQL DB
- Provides access role of the customer (e.g. user, admin)

5.2 Scenarios

The scenarios are a sequence of logical steps in order to use the Panna product. It provides some technical explanation but not in great detail. The technical details are

covert in the next section. The Service column provides a quick overview of the required services.

Services	Scene
Browser Pana Website	**First Contact** A customer visits the website of the Panna project. Information about the project can be found there as well as a block and/or community for recent updates, bug fixes and feedbacks. Moreover, one doesn't need to register to download the software. [Technical information are missing!]
Registration Manager	**Registration** After the software is downloaded and installed one need to register at the backend of the system. This is done by Registration Manager services. In order to do that one has to create an account with certain information, namely: name, surname, email and password. The Registration Manager service, check the validity of the REST message in order to avoid corruption of the messag. After validation, the Registration Manager approves the user, it will return a positive boolean value. This allows the Credential service to generation a token by means of SSO (Single-Sign on), which will be returned to the Application Panna Client. It is now able to use the session to communicate with the backend without to login all the time. The token has also a lifespan, it expires when the user logs out. In case the user forgets to logout, then the token expires after 30min [Time was chosen randomly!]. Finally, the Registration Manager insert, updates or deletes the user credentials into the SQL DB. Further, the Registration Manager service provides all users with certain rules (e.g. user, admin). The rules and tokens can be read by the other services within the backend environment via repository services.
Pana Application	**How data en- and decryption work (Frontend)** Now the software is installed and the user is logged in. Hence, a user wants to use the product. [How will it work?] <Andreas?>

Agent Data Manager	**How storing of encrypted data work (Backend)** The data arrives at the Agent service encrypted, thus the Agent knows nothing about the data. Therefore, there are two security checks for the agent to do: namely integrity and authentication. The integrity of the cipher data is done by means of a checksums. Anyway, both security criterias can be managed by WS-* security standards. After checking the encrypted data, the Agent passes it to the Data Manager by means of a well defined API. The Data Manager parses the incoming data and distracts the meta information (storing location of the data) from the actual data. The meta information can be passed via the SOAP HEADER. If need be, the SOAP message can be encrypted too. Moreover, the Data Manager add mapping information to the metadata in order to find it again in the future. Furthermore, the service task is to insert, update or delete the metadata in the Storage System [What kind of storage system?], which represent the real location of the data.
???	**Possible security threats** **(not only malicious but also accidentally once)** It is advisable to read the third chapter: “Sicherheitsaspekte Service-orientierter Architekturen” in the paper: “SOA-Security-Kompendium” [4]. Unfortunately, the paper is written in german, thus it needs to be translated for a not german speaker. Nevertheless, the third chapter in the paper contains general security risks for SOA environments. This information should make one more coutious in order to develop the system, as well as to find out project specific security risks. [What are the potential threats to the program?] Consider the following: (see also Glossary) ● Authentication ● Authorization ● Confidentiality ● Integrity ● Availability ● Accounting ● Auditing

???	**Payment System** //TODO: Continuing after foundation is built, tested and released (beta).
???	**Third Party Storage Feature** //TODO: Continuing after foundation is built, tested and released (beta).
???	**Image Sorting and Tagging Feature** //TODO: Continuing after foundation is built, tested and released (beta).
???	**Data Sharing Feature** //TODO: Continuing after foundation is built, tested and released (beta).
???	**Current state of the data, as well as their location needs to be backuped, by means of an lightweight DB.** //TODO: Continuing after foundation is built, tested and released (beta).

6 REFERENCES

[1]	WhatsApp (April, 2016): “WhatsApp Encryption Overview” Publisher: WhatsApp; Accessed: 09/April/16 https://www.whatsapp.com/security/
[2]	Dominik Schürmann (April, 2016): “open-keychain/open-keychain” Publisher: © 2016 GitHub, Inc.; Accessed: 09/April/2016 https://github.com/open-keychain/open-keychain
[3]	Threema GmbH (March, 2016): “threema-ch” Published: © 2016 GitHub, Inc.; Accessed: 13/April/2016 https://github.com/threema-ch
[4]	Bundesamt für Sicherheit in der Informationstechnik (2009): “SOA-Security-Kompendium” Publisher: © Bundesamt für Sicherheit in der Informationstechnik 2009; Accessed: 21/April/2016; URL: https://www.bsi.bund.de/EN/Topics/OtherTopics/SOAsecurity/Basics/Examplescenarios/examplescenarios_node.html Paper (German): https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/SOA/SOA-Security-Kompendium_pdf.pdf?__blob=publicationFile&v=1