myVerein

Software Requirement Specification

Student Research Project

for the

Bachelor of Science

at Course of Studies Applied Computer Science
at the Cooperative State University Stuttgart

by

Frank Steiler

September 2014

Time of Project	13 Weeks
Student ID, Course	8216767, TINF12A
Company	Hewlett-Packard GmbH, Böblingen
Supervisor	Alfred Becker

Contents

Chapter 1

Product purpose

The system is intended to simplify the management of a club by unifying the communication channel between the representatives of the organisation and the members. This goal is going to be achieved by using modern technology. The product should be implemented as a server client architecture, where the client is an iOS running smartphone or tablet.

1.1 Obligatory requirements

The fulfilment of the following criteria is mandatory:

User

- The user needs to be able to log himself into the server provided by his club at the start of the application
- The user needs to be able to check the upcoming schedule of the club
- The user needs to be able to send and receive messages

Administrator

- The administrator needs to be able to modify the server according to the clubs name etc.
- The administrator needs to be able to modify the access rights of all members of the club
- The administrator needs to be able to schedule an event

Application (Client)

- The application needs to be optimised for the operation with an iPhone 6
- The application needs to ensure an intuitive operation
- The application needs to have a logic menu structure
- The application needs a basic graphical user interface

System (Server)

- The system needs to ensure a fault tolerant, consistent operation.
- The system needs to be configurable
- The system needs to provide a secure user authentication
- The system needs to handle the login of multiple users at the same time
- The system needs to handle several messages at the same time
- The system should be configurable through a web interface, that enables the administrator to manage user
- The system needs to operate according to the data privacy act
- The system needs to be designed in a way that a user can only access a minimum amount of private data of the other users
- The system needs to be designed to be easily extensible
- The system needs to gather the data from a relational database
- The system needs to be developed in Java 8

1.2 Optional requirements

The following requirements are optional, but their implementation is nice to have.

- The application should have a sophisticated graphical user interface
- The administrator should be able to create divisions

- Each user should be able to be part of one or more divisions, to only receive relevant information.
- The system should support private chats for each division
- Invitations to events should be send to divisions
- The user should be able to request access to a division, this access is granted by a higher level user
- The user should be able to share photos that are relevant to the club
- The application should use push notifications to effectively alert the user about incoming messages, news or upcoming events
- The system should be designed to ensure extensibility
- The application should be developed using Swift
- The user should be able to optional create a public profile containing contact information
- The application should be created according to the Apple Developer Guidelines

1.3 Additional requirements

The following requirements would improve the overall user experience, but their implementation is not business critical.

- The application could have a central news feed, which contains the latest information provided by the administrator
- The administrator could be able to publish relevant information through the web interface
- The application could be implemented multilingual
- The system could have a second member type, that is allowed to publish news and events
- The application could be able to send and receive attachments in messages, like photos or audio messages

- The application could handle multiple accounts on different or on the same server
- The application could support the native resolution of different devices, like the iPad or the iPad mini
- The system could send an Email newsletter for members that are not owning a smartphone running iOS
- The events could support assignment of supporting roles needed during the event
- The events could support voting buttons to find the ideal date for the meeting
- The system could support the download of shared pictures
- The system could support the use of shared pictures within the news of the club
- The application could support user management for administrating users within the application

1.4 Non-requirements

The following requirements are not in the scope of this product.

- The system is not designed to create a homepage for the club
- The application is not designed to provide access to non-members of a club
- The application is not designed to work without a server where the user is able to authenticate himself
- The system should not be used to collect statistics about the users

Chapter 2

Product environment

2.1 Application area

The system combining a messaging app, as well as event organisation and news publishing service. These functionalities are intended to be used in the context of a single club or organisation.

2.2 User group

The application can be used by every registered member of an organisation, owning a smartphone with internet connection. Concluding the application is going to be used by a wide age group, including people that are not very familiar with the technology they are operating.

The setup and administration of the server environment should be done by a person that is a skilled IT administrator.

2.3 Operation condition

The application is intended to run on a iPhone 6 in upright orientation, operated by one user at a time. The server application needs to run on a web server with any operating system.

Chapter 3

Technical requirements

3.1 Software

The users device needs to run at least iOS 8. The server is recommended to run a current version of Linux, for example Debian Whezzy or Ubuntu 14.04 LTS.

3.2 Hardware

To use the application the user needs an iPhone or iPad of an independent revision. The server needs to be scaled according to the amount of registered members and requests.

Chapter 4

Product functions

<PF_A_0000> Launch application
<PF_A_0010> Show splash screen
<PF_A_0011> Show login form
<PF_A_0020> Log into the system using user credentials and server address
<PF_A_0021> Request access to a server
<PF_A_0030> View schedule of the club
<PF_A_0040> View detailed information about upcoming events
<PF_A_0050> View list of all conversations the user is a part of
<PF_A_0060> Select and view a conversation the user is a part of
<PF_A_0070> Send a message to a conversation the user is a part of
<PF_A_0080> Receive a message within a conversation the user is a part of
<PF_A_0090> Request access to additional divisions
<PF_A_0100> Share photos by uploading them to the server
<PF_A_0101> Change private information (Email address, Password, etc.)
<PF_A_0102> Reset password using provided email
<PF_A_0110> Create public profile
<PF_A_0120> Edit public profile
<PF_A_0130> View public profile of other user
<PF_A_0140> Receive push notifications
<PF_A_0150> Log out of the system
<PF_A_0160> Quit application
<PF_A_0170> Pause application

<PF_S_0010> Launch server application

<PF_S_0020> Login of administrating users only
<PF_S_0030> Initial setup of application through user interface
<PF_S_0040> Create new user
<PF_S_0050> Delete existing user
<PF_S_0060> Change membership of user within divisions
<PF_S_0070> Create new event
<PF_S_0080> Add division to event
<PF_S_0090> Change event
<PF_S_0100> Authenticate user
<PF_S_0110> Receive, process and relay messages of authenticated users
<PF_S_0120> Change administrator credentials
<PF_S_0130> Edit administrating user
<PF_S_0140> Securely connect to database and gather relevant information
<PF_S_0150> Log out of system

Chapter 5

Product data

5.1 Non-persistent data

<PD_A_0000> Viewed user profiles

<PD_S_0000> Currently logged in administration user

5.2 Persistent data

<PD_A_0010> Login credentials (Until log out)
<PD_A_0020> User settings
<PD_A_0030> Received messages for a user specific amount of time (Default 1 month)
<PD_A_0040> Scheduled events for a user specific amount of time after the event occurred (Default 1 month)
<PD_A_0050> User information

<PD_S_0010> User information, encrypted credentials and their affiliations within the club
<PD_S_0020> Non delivered messages
<PD_S_0030> All upcoming events and their invited divisions
<PD_S_0040> All uploaded user photos

Chapter 6

Quality requirements

6.1 User application

	Very important	Important	Normal	Un-important
Reliability	x			
User friendly	x			
Efficiently		x		
Functionality		x		
Portability			x	
Adjustability			x	

Table 2: Quality requirements for the user application

Reliability (Correctness & fault tolerance): It is very important for the user application to run seamlessly and does not crash unexpectedly.

User friendly: It is very important to create a user friendly application, whose interface is intuitively usable. A user should not need a special training to be able to operate the application.

Efficiently: It is important to establish an efficient workflow to ensure a frustration free operation.

Functionality: It is important, that the application is meeting all obligatory and optional requirements.

Portability: The app should primary run on Apple smartphones and the porting to an Apple tablet or Android device is optional.

Adjustability: The core of the application is not intended to be massively altered after the initial implementation.

6.2 Server application

	Very important	Important	Normal	Un-important
Reliability	x			
User friendly			x	
Efficiently		x		
Functionality		x		
Portability	x			
Adjustability			x	

Table 3: Quality requirements for the server application

Reliability (Correctness & fault tolerance): It is very important for the server application to run seamlessly and does not crash unexpectedly.

User friendly: A user friendly interface is not very important since the system application should only be operated by skilled administrators.

Efficiently: It is important to establish an efficient workflow to ensure a frustration free operation.

Functionality: It is important, that the application is meeting all obligatory and optional requirements.

Portability: The application needs to support a wide array of operating systems and thereby it is very important to have a high portability.

Adjustability: The core of the application is not intended to be massively altered after the initial implementation.

Chapter 7

Test cases

7.1 User application

Table 4: User application test cases

Identifier	Description	Expected result
<PT_A_0000>	Launch application • <PF_A_0000> • <PF_A_0010> • <PF_A_0011>	The Application launches without any error
<PT_A_0010>	Planned stop of the application • <PF_A_0160>	The application is stopped without the loss of data
<PT_A_0020>	Unplanned stop of the application, e.g. empty battery • <PF_A_0160>	The application is stopped without the loss of data

7.2 Server application

Table 5: Server application test cases

Identifier	Description	Expected result
<PT_S_0000>	Launch application • <PF_S_0010>	The Application launches without any error
<PT_S_0010>	Unplanned stop of the application, e.g. server reboot	The application does not loose any data

Chapter 8

Non-functional requirements

Since the Software uses sensible user information, it has to fulfil the requirements of the corresponding laws in the country where it should be used. In this special case, it has to fulfil the Bundesdatenschutzgesetz. Therefore it is not allowed to store any data, which is not directly needed for the system. If there are additional Information stored, the user has to be informed and needs to agree to that usage.

The system should consume as less CPU – Power as possible, to make the complete server as efficient as possible, while still ensuring a reliable operation. This can be achieved by efficient programming and a good initial design.

The complete source code is licensed using a GNU GPL version 2 license.

Chapter 9

User interface

The user interface is going to use a common color scheme: `http://paletton.com/#uid=73i0u0kllll97A-ferBrsf4-F8P`

9.1 User application

The application should have a tabbed layout (Like the Facebook app). There would be 4-5 entries, depending on the features that are going to be implemented within the system: Calendar, Chats, Photos, News, Settings.

9.2 Server application

For the design of the web interface the Twitter Bootstrap engine is going to be used.

Chapter 10

Development environment

10.1 Software

For the development of the iPhone application the xCode IDE is going to be used, while the Java based server application is developed using Netbeans IDE.

10.2 Hardware

The software is going to be developed on a MacBook Pro Mid 2010, 15", while the client application is tested on an iPhone 6. The server application is going to be tested on **???**.

Chapter 11

Key milestones

The milestones given in this document are just rough estimators.

Table 6: Key milestones and their scheduling

#	Milestone	Target start date	Target completion date
1	Software Requirement Specification	17.10.2014	20.10.2014
2	Server Specification	20.10.2014	3.11.2014
3	Basic Server implementation (Obligatory requirements)	03.11.2014	24.11.2014
4	User Application Specification	20.10.2014	05.01.2015
5	Basic User Application Implementation	05.01.2015	02.02.2015
6	Bug fixing and implementation of interaction between client and server	02.02.2015	23.02.2015
7	Implementation of Optional requirements	23.02.2015	13.04.2015
8	Writing of project paper	17.10.2014	20.04.2015
9	Revision of paper and further bug fixing	20.04.2015	18.05.2015

Continued on next page

Table 6 – continued from previous page

#	Milestone	Target start date	Target completion date
10	Implementation of additional requirements	13.04.2014	...

List of Figures

List of Tables