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 bilingual (English, German)
- 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 is 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

Product function codes with an "A" in the middle are refering to the client application, respectively "S" is refering to the server application.

<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_0011> Access server user interface through web browser
<PF_S_0012> Show login form
<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_0041> Accept user request
<PF_S_0042> Decline user request
<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_0111> Receive and process uploaded pictures of authenticated users
<PF_S_0120> Change administrator credentials
<PF_S_0130> Securely connect to database and gather relevant information
<PF_S_0140> Log out of system

Chapter 5

Product data

Product data codes with an "A" in the middle are refering to the client application, respectively "S" is refering to the server application.

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> Upcoming events and their invited divisions

<PD_S_0040> Uploaded user photos
<PD_S_0050> The club and his metadata (Club name, Register number, head of administration)

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 1: 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 2: 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 3: 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

Continued on next page

Table 3 – continued from previous page

Identifier	Description	Expected result
<PT_A_0030>	Interruption through system events (e.g. phone call)	The application pauses without the loss of data and relaunches at the last view
<PT_A_0040>	Log into the application • <PD_A_0011> • <PF_A_0020>	The application loads all information provided by the server
<PT_A_0050>	The user requests access to the server, providing his name and email address • <PF_A_0021>	The application reports a successfull request and the user receives an email notification
<PT_A_0060>	The user receives access to the server • <PF_S_0041> • <PD_A_0011> • <PF_A_0020>	The user receives an email notification and is able to log into the application
<PT_A_0070>	The user has forgotten his credentials and needs to reset his password from the login screen • <PD_A_0011> • <PF_A_0102>	The user receives an email notification with his new credentials

Continued on next page

Table 3 – continued from previous page

Identifier	Description	Expected result
<PT_A_0080>	The application syncs the data from the server • <PF_S_0140> • <PF_A_0080>	The application connects to the server and receives only relevant and changed information (E.g. new/changed events or new messages)
<PT_A_0090>	Select calendar menu entry • <PF_A_0030>	The application shows a calendar view with all upcoming events, published by the club and relevant to the user
<PT_A_0100>	Select calendar event • <PF_A_0040>	The application shows detailed information about the selected event
<PT_A_0110>	Select messages menu entry • <PF_A_0050>	The application shows list of all user relevant conversations
<PT_A_0120>	Select a message • <PF_A_0060>	The application shows list of all user relevant conversations
<PT_A_0130>	Send a message • <PF_A_0070> • <PF_S_0110>	The application successfully sends the message to the server and then relays it to the correct receipients

Continued on next page

Table 3 – continued from previous page

Identifier	Description	Expected result
<PT_A_0140>	Select divisions menu entry • <PF_A_0090>	Present a list of all divisions of the club
<PT_A_0150>	Press on the request access to division button • <PF_A_0090>	The application requests access to a certain division, the responsible administrator gets notified and the user is informed about the successful request
<PT_A_0160>	Select photo menu entry • <PF_A_0100>	The application shows a camera view, with the option to choose an existing photo
<PT_A_0170>	Upload a selected photo • <PF_A_0100> • <PF_S_0111>	The application uploads the photo, the server stores the photo and the application notifies the user about the successful upload
<PT_A_0180>	Select the settings menu entry	The application presents a list of available settings options
<PT_A_0190>	Select the public profile menu entry • <PF_A_0110> • <PF_A_0120>	The application shows the current public profile of the user and the user is able to edit the entries

Continued on next page

Table 3 – continued from previous page

Identifier	Description	Expected result
<PT_A_0200>	Save the public profile • <PF_A_0110> • <PF_A_0120>	The application uploads the updated public profile of the user to the server
<PT_A_0210>	Select a contact within a conversation • <PF_A_0060> • <PF_A_0130>	A detailed view of all private information shared by the contact is displayed
<PT_A_0220>	The application receives push notifications • <PF_A_0140>	Depending on the users settings he receives a push notification for an event, e.g. an incoming message or upcoming event
<PT_A_0230>	Select the logout menu entry • <PF_A_0150> • <PF_A_0011>	The application logs out of the system and presents the login screen

7.2 Server application

Table 4: 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
<PT_S_0020>	Access admin web page for the first time • <PF_S_0030>	The server presents the process of the initial setup
<PT_S_0030>	Access admin web page • <PF_S_0011> • <PF_S_0012>	The server presents a login window
<PT_S_0040>	Log into the system using credentials • <PF_S_0020>	The server proceeds to the administrator view if the credentials match to an administrator account
<PT_S_0050>	Create new user • <PF_S_0040>	The provided name and email address is stored and the user receives an email containing his credentials
<PT_S_0060>	Select “access requests” menu item	The server provides a list of users that requested access to the server

Continued on next page

Table 4 – continued from previous page

Identifier	Description	Expected result
<PT_S_0070>	Process access request • <PF_S_0041> • <PF_S_0042>	The provided name and email address is stored and the user receives an email containing his credentials
<PT_S_0080>	Select "user list" menu item	The server provides a list of all registered users
<PT_S_0090>	Delete user • <PF_S_0050>	Selected user is deleted and is not able to log himself into the system anymore
<PT_S_0100>	Reset user password	The user is notified about his new password through email
<PT_S_0110>	Change membership within divisions • <PF_S_0060>	The user is added or removed from the division, this change is respected in the following updates of the user
<PT_S_0120>	Select events menu item	A calendar view with all upcoming events is presented
<PT_S_0130>	Select event • <PF_S_0080> • <PF_S_0090>	The user is seeing all details of the event and is able to change the invited groups as well as change the event itself

Continued on next page

Table 4 – continued from previous page

Identifier	Description	Expected result
<PT_S_0140>	Saving selected element • <PF_S_0080> • <PF_S_0090>	The event is changed and the changes are pushed to the user
<PT_S_0150>	A users try to log into the system • <PF_S_0100> • <PF_A_0020>	The server checks if the user is allowed to log in and sends back the response
<PT_S_0160>	A message is sent to the server • <PF_S_0110>	The sender's authentication is checked and the messages are queued until the recipient are online
<PT_S_0170>	A photo is uploaded to the server • <PF_S_0111>	The photo is stored on the server
<PT_S_0180>	Select settings menu item	The administrator is presented a list of adjustable settings
<PT_S_0190>	Change administrator password • <PF_S_0120>	The new password is stored within the system and the administrator is only able to log into the system using his new password

Continued on next page

Table 4 – continued from previous page

Identifier	Description	Expected result
<PT_S_0190>	Select log out menu item • <PF_S_0140>	The user is logged out of the system and the server shows the log in page

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 and the documentation is going to be licensed using a GNU GPL version 2 license. If the content is not applicable to the GNU GPL version 2 license a Creative Commons - Attribution - Non Commercial - Share Alike - 4.0 International License will be used.

Chapter 9

User interface

For the user interface a common color scheme needs to be evaluated. This scheme would help the user to associate a logo, web page or user interface to an application or brand.

9.1 User application

The application should have a tabbed layout (Like the Facebook app). This view would have up to 5 entries: Calendar, Chats, Photos, News, Settings.

9.2 Server application

The responsive Twitter bootstrap engine would be used to design the user interface for the web application administrating the server application.

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". The client application is tested on an iPhone 6, while the server application is going to be tested on a root server provided by the university running a common Linux distribution.

Chapter 11

Key milestones

The milestones given in this document are just rough estimators.

Table 5: 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	Interaction between client and server implemented & system bug free	02.02.2015	23.02.2015
7	Optional Requirements Implementation	23.02.2015	13.04.2015
8	Project Paper	17.10.2014	20.04.2015
9	Revised project paper, applications bug free	20.04.2015	18.05.2015

Continued on next page

Table 5 – continued from previous page

#	Milestone	Target start date	Target completion date
10	Additional Requirements Implementation	13.04.2014	08.06.2015

List of Figures

List of Tables