



ET1446: SOFTWARE DEVELOPMENT FOR TELECOMMUNICATION SYSTEMS

*This document is presenting all test specifications as also defined in design document.*

## ACCEPTANCE TEST PLAN V1.1

Group: SWAT Kats

Sokratis Papadopoulos	Jibraan Singh Chahal
Johannes Grohmann	Jyoti
David Alarcón Prada	Akanksha Gupta
Gautam Vij	Rohit Raghav
Inanc Gurkan	

# ACCEPTANCE TEST PLAN

## 1. PREFACE

This document is about the test units that will check the functionalities of our program “**LetsTalk**”.

In the following we will first provide a glossary with a list of abbreviation in chapter 2 and then we present all tests defined by the design document.

### VERSION HISTORY OF THIS DOCUMENT:

#### ***Release v1.1 on 2015-05-14***

- Added test cases for remove, block and unblock in section 3.3.

#### ***Release v1.0 on 2015-05-12***

- Initial release

## 2. GLOSSARY AND ABBREVIATIONS

In this section are defined technical terms that are in this document in alphabetical order.

- **ADDBOOK:** address book
- **PWD:** password
- **REQ:** requirement

## 3. ACCEPTANCE TEST PLAN

### 3.1 LOGIN

**Test:** TEST\_LOGIN

**Purpose:** It will test that the login authentication works properly

**Requirements:** REQ\_USR\_1

**Environment:** executed the application (login windows appeared)

**Operation:** run the executable providing as parameters a valid username-password. Also run it providing a false username & password.

**Expected result:** login for authorized users / failure message for non-authorized

**Result:** success/failure

### 3.2 SETTINGS

**Test:** TEST\_SETTINGS\_STATUS

**Purpose:** It will test that the changing of user's status functions properly

**Requirements:** REQ\_USR\_3

**Environment:** user must be logged in our application

**Operation:** run the executable providing as parameters a valid username-password pair and the desired status (Available/Busy).

**Expected result:** status is changed/set according to user's desire

**Result:** success/failure

**Test:** TEST\_SETTINGS\_CHANGE\_PWD

**Purpose:** It will test that the changing of user's password functions properly

**Requirements:** REQ\_USR\_4

**Environment:** user must be logged in our application

**Operation:** run the executable providing as parameters a valid username-password pair and the desired new password.

**Expected result:** password is changed according to user's desire

**Result:** success/failure

### 3.3 ADDRESS BOOK

**Test:** TEST\_ADDUSR

**Purpose:** Registered user must be able to add other registered users in his address book by selecting users from global address book.

**Requirements:** REQ\_USR\_2

**Environment:** user must be logged in our application

**Operation:** : run the executable providing parameter as a valid username from global address book and choosing an option of add.

**Result:** success/failure

**Test:** TEST\_REMVUSR

**Purpose:** Registered user must be able to remove other registered users from his local address book.

**Requirements:** REQ\_USR\_4

**Environment:** user must be logged in our application and user exists in local address book.

**Operation:** run the executable providing parameter as a valid username from local address book and choosing an option of remove.

**Expected result:** If a contact is removed, then it should not appear in user's address book.

**Result:** success/failure

**Test:** TEST\_BLKUSR

**Purpose:** Registered user must be able to block users in his address book

**Requirements:** REQ\_USR\_3

**Environment:** user must be logged in our application

**Operation:** run the executable providing parameter as a valid username and choosing an option of block.

**Expected result:** If a contact is blocked then it should not appear in local address book (if he was added first) and they should not be able to chat with each other.

**Result:** success/failure

**Test:** TEST\_UNBLCKUSR

**Purpose:** Registered user must be able to unblock users from his block list.

**Requirements:** REQ\_USR\_5

**Environment:** user must be logged in our application and user exists in blocked list.

**Operation:** run the executable providing parameter as a valid username from blocking list and choosing an option of unblock.

**Expected result:** If a contact is unblocked then it should not appear in address book and they should be able to chat with each other (If any of them adds another one).

**Result:** success/failure

### 3.4 TEXTING

**Test:** TEST\_MESSAGE

**Purpose:** Successfully transmit a message from the sender to the receiver.

**Requirements:** REQ\_USR\_5

**Environment:** Both sender and receiver have to be online and exchanged keys with the server.

**Operation:** Transmit the message to the required receiver.

**Expected result:** The message has to be received by the receiver. The message shall be the same, as the sender sent it, e.g. no artifact through encryption should occur. If the receiver is an invalid user an exception or a notice should be thrown and the message should not be transmitted. No other users except the specified receiver should get the message.

**Result:** success/failure

### 3.5 FILE TRANSFER AND MATERIAL REPOSITORY

**Test:** TEST\_FILE\_TRANSFER

**Purpose:** System shall allow a user to send files

**Requirements:** REQ\_USR\_5

**Environment:** User must be logged in our application

**Operation:** Run the executable providing as parameters a valid username-password pair, file name and the receiver.

**Expected result:** File is received by the receiver and is able to download the file

**Result:** success/failure

### 3.6 ADMIN TOOLS

**Test:** TEST\_ADMIN\_TOOLS

**Purpose:** Add/remove users globally, view the statistics of the chats, or make another user admin

**Requirements:** REQ\_USER\_6

**Environment:** (Admin) User must be logged in our application

**Operation:** Run the executable providing as parameters a valid username-password pair

**Expected result:** Admin can view the conversation call records. If a user is added globally, other users can add the new user in their local address book. If a user is deleted globally, it should be removed from the local address books of every user that has previously added the deleted user. Admin can see a general graph of all the chat activities in LetsTalk

**Result:** success/failure