

UNIVERSITY OF CAMBRIDGE

COMPUTER SCIENCE TRIPOS, PART IB

GROUP PROJECT TEAM ECHO

---

# Multi-touch Conference Functional specification

---

*Authors:*

Mona Niknafs (mn407)

Yojan Patel (yp242)

Alexandru Tache (at628)

Philip Thomson (prt28)

Petar Veličković (pv273)

*Client:*

Catherine White, BT

28 January 2014



UNIVERSITY OF  
CAMBRIDGE

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Purpose . . . . .	1
1.2	Scope . . . . .	1
1.3	Definitions . . . . .	1
1.4	References . . . . .	1
1.5	Overview of the remainder . . . . .	1
<b>2</b>	<b>Overall Description</b>	<b>2</b>
2.1	Product perspective . . . . .	2
2.2	Product functions . . . . .	3
2.3	Use cases . . . . .	3
2.4	Facilities / environments required . . . . .	4
2.5	User characteristics . . . . .	5
2.6	Constraints . . . . .	5
2.7	Assumptions and dependencies . . . . .	5
2.8	Acceptance criteria . . . . .	6
<b>A</b>	<b>Management Strategy</b>	<b>7</b>
<b>B</b>	<b>Further Work</b>	<b>8</b>

# 1 Introduction

## 1.1 Purpose

The purpose of this document is to provide a complete description of the Multi-touch Conference system. The functionality of each module of the system will be presented, with respect to their interactions with internal and external elements.

This document is intended for our client, group project coordinators and system developers.

## 1.2 Scope

The Multi-touch Conference system will be a conference tool which allows users to take part in one or more collaborations between other users of the system (locally and remotely), with or without a multi-touch screen.

The system will optimise ease of starting and migrating between conversations of interest. This will be achieved through keyword recognition in conference conversations, to well-define the subject of each conversation. User profiles will be used to keep track of important user data such as interests and conversation history, to the benefit of both the user and the conference system as a whole.

An external smartphone and tablet application will also be developed as a companion to the system, which will interact with the touchscreen and chat server to allow individual users to chat in conferences. These components will be integrated with a simple user interface on the display to each circle of local conference members, where touch screen gestures will allow for a more collaborative environment.

## 1.3 Definitions

Term	Definition
Admin	User with the ability of starting up a server and/or starting conferences.
Conference	Collection of several conversations between users with shared interests.
Conversation	Exchange of messages between two or more users within a conference.
Database	Storage medium for the information collected by the system.
Real-life networking	Sharing information and services between individuals sharing a common interest.
User	System account holder, attendee of conferences.
User profile	Unique description of a particular user, through activity and preferences.

## 1.4 References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998.

## 1.5 Overview of the remainder

The remainder of this specification begins with an overall description of the system. This section will give a universal description of the product functions with reference to facility requirements and user characteristics.

This is followed by the specific requirements section, a more detailed description of the constraints that will likely be encountered with regards to the system functionality.

The appendices at the end of the document explain our chosen management strategy and our proposals for further work.

## 2 Overall Description

### 2.1 Product perspective

The current web conferencing market provides facilities for remote collaboration between two people. This does not easily extend to allow for networking between groups of people, which is more closely analogous to real world conferencing.

The key feature of this real world system is the ability to move between established conversations, with a vague knowledge as to which conversations are available. There currently exist some close solutions to the problem in the market today, however none completely encapsulate this key feature.

*Webex*, an established software provider in the one-to-one web conferencing market, provides conferencing through features such as desktop sharing, webcams, phone conferencing and user profiles. While the desktop sharing is a useful tool for education and training, it could be argued to be impractical; parties involved in remote collaboration must crowd around a desktop intended for one operator. A powerful feature of Webex are the user profiles which increase the ease of starting impromptu conversations; a property which is necessary in the conference system.

Another system which accommodates web conferencing is *TeamViewer*. It provides desktop sharing tools with a remote desktop control facility. The TeamViewer software provides a quick and reliable UDP/TCP connection and communication is authenticated via an ID and password combination. The TeamViewer server handles all user traffic. This is a very well designed system, but the design goal of TeamViewer is not to provide a collaboration tool analogous to real-world conferencing. Therefore communication between users during sharing is difficult and it is not very easy to migrate between conversations.

*DiamondTouch* is a conferencing tool, which unlike TeamViewer and Webex, uses multi-touch single-display collaborative software. A useful feature provided is user recognition, where multiple users using the display are each recognised via a transmitter array located in the touch surface and the chairs of each user. Although a distinguishing feature, it is arguably complex, perhaps user recognition can be made a more simple process via choice of user interface design. Additionally, it seems too large a constraint to require users of a display to be sitting down to give correct functionality.

A system similarly combining multi-touch hardware and software is *Microsoft PixelSense*. The product uses near-infrared cameras to detect both fingers and objects placed on the display. Objects of a specific size and shape placed on the display result in a preprogrammed response by the software. Microsoft PixelSense is designed for commercial use in public places, without the need for a keyboard or mouse. The system uses Windows 7 for Embedded Systems. Although the design goals are different to that of a conference system, the multi-touch hardware and software combination used by Microsoft PixelSense can be adapted to apply to the Multi-touch Conference system.

Considering what is currently in the market today, there exist some features on similar systems which can be translated to the Multi-touch Conference system. User profiles will increase ease of beginning impromptu conferences, as more information is known about the users using the network. Furthermore, an internal server seems to be an efficient way of dealing with user traffic. User recognition is a necessary feature with a multi-touch display; a well designed user interface could handle this though location detection.

## 2.2 Product functions

This product is designed to make a conference over a network connection or even the Internet to allow conferencing to take place without the cost and complexities of organising and holding physical conferences. Also it will allow viewing and analysis of the data collected during the conference (and possibly during previous conferences).

As a conferencing software this product will have one main division between the clients (in this section (2.2), 'client' is used to refer to the user's local program) and the server.

The client needs to handle:

- displaying information from both the messages/chat and the statistics;
- the user input ranging from sending messages to changing conversations and requesting statistics;
- establishing a secure and robust connection with the server.

The server needs to handle:

- the data aggregation and analysis/ranking;
- the client requests;
- the management of conversations and participating clients;
- creating a secure and robust connection between itself and the many other clients.

As part of this major division, messages (packets of data arranged so that both sides can create and use the data to complete the functions) can be used to move data between sections of this product. These messages will restrict the type of data that can be transferred between the server and the clients. The product will work by the user's local client receiving a message from the user via a form of input device (keyboard, touchscreen. . . ) then packaging the message into a computer readable message object which is then sent to the server. The server then handles multiplexing of the messages depending on their type, forwarding them to the appropriate processing class within the server. Afterwards the server will send a reply to the requesting client, and any other clients that may be affected.

Using the server software, the user can make a conference that other people with the appropriate client software can attach to. The server will account for most of the processing and analysis of the data and the collection of the data. It will contain the functions to save conversations and restore if there is an issue, and it also needs to handle maintaining conversations.

With the standard client a user can connect to the server. The user can send messages, receive messages, join/create conversations with groups of other users and view statistics about the conference. All of these functions are computed mainly in the server as lots of the information needed resides there and to move computation away from the client which could be running f.ex. on a comparatively slow Android device.

A touchscreen client gives the opportunity for multiple attendees in the same room to have a communal discussion viewer and added functionally for swapping conversations and statistics viewing. This is based on the normal client but does not have the message sending ability.

With the touchscreen program providing a helpful user interface for a group of people all attaching to the same conference from the same room, there is a need for the android app to communicate with this program. This product is going to use QR codes to allow the multi-touch screen program to give information to the Android/smartphone app. This QR code then will allow the app to send the relevant request to the server automatically. This means the app can scan a QR code and then use that to attach to a conversation quickly and easily. This requires the touch-screen to be able to generate the QR codes and the android device to be able to read and interpret the QR codes.

## 2.3 Use cases

- Administrator and server:
  1. Administrator decides that he/she needs to make a conference.

2. Administrator starts the software and decides on the port number and conference name (and possibly password).
  3. Administrator disseminates these details to attendees along with time of the conference by a method such as e-mail.
  4. Administrator checks that the server is running when the conference is due to start.
  5. The attendees all attach and the conference starts.
  6. Once the conference finishes the administrator takes the server down and saves the data for analysis and referral in the future.
- Attendee and client (local program):
    1. The attendee receives an invitation to a conference from the conference administrator containing key information about the conference including how to login and when it is to run.
    2. When the time comes to attend the conference, the attendee uses his/her client software on his/her pc/smartphone/tablet to connect to it.
    3. The attendee searches through the conversations until he/she finds one that he/she likes and joins and converses.
    4. After a few minutes he/she remembers that there was an important project that he/she needs to discuss and searches for a conversation on it but doesn't find it.
    5. He/she creates a conversation about that project including keywords on the project as tags.
    6. The other contributors to that project see the creation of the conversation and join and then discuss the project.
    7. The attendee exits the conversation when it is finished and exits the client program.
  - Attendee and client (local program) with multi-touch screen:
    1. The attendee receives an invitation to a conference from the conference administrator containing key information about the conference including how to login and when it is to run.
    2. When it is time to attend the conference, the attendee goes to a conference room with fellow attendees who are in the same building and uses his/her client software on his/her smartphone/tablet.
    3. The client searches through the conversations until he/she finds one that he/she likes and joins and converses.
    4. On the touch screen he/she sees an interesting conversation going on at the moment and so uses his/her application to scan a QR code that will be associated with the conversation.
    5. The app then takes the attendee to that conversation where he/she joins the conversation and gives her/his input on the conversation.
    6. Then the attendee looks at the statistics displayed on the touch screen seeing that a very popular conversation is something that he/she wanted to talk about.
    7. The attendee then searches for the name of that conversation and finds it; then he/she joins it and converses.

## 2.4 Facilities / environments required

The facilities that will be required for this software are a computer for the server program to run on and a combination of computers or other devices such as smartphones or tablets for the conference attendee program.

The computer that runs the server needs to be sufficiently powerful, depending on the number of attendees expected for the conference(s) it supports. The server will need to have reliable and fast enough connections to service all the attendees.

The clients will need supported devices such as a PC or a smartphone with networked connection to the server fast enough for simple message passing. The attendees' devices can be much less powerful than the server as the client program will be able to be run on low power devices. To use the multi-user touchscreen software a multi-touch screen and a computer with enough power to use said screen and networked connection to the server would be required.

## 2.5 User characteristics

There are two main users of the system: the conference creator (who may him/herself be an attendee) and the conference attendees. Both users have very different needs and different programs to achieve this and may need to use different platforms or devices.

The conference creator needs a program that will create the server for others to attach to and store the data. This will need to be a larger program with data analysis and data storage allowing the conference to happen, and will need to be permanently hosted on a reasonably powerful computer.

The conference attendees will need a program that allows for joining the conference, joining conversations, conversing and viewing the statistics. This program or versions of it will need to run on very low powered devices such as smartphones. Some conference attendees may be in small groups and as a group use the same program on a large touch screen to allow for multiple attendees to attach at once. The attendees will need no particular training in order to use this application, as it will be designed with simplicity and user-friendliness in mind.

## 2.6 Constraints

A touchscreen interface that can handle 10 touches is used for the main user interface. This constrains the design of the user interface to discourage a situation where more than the allowed amount of touches occur. This also constrains the UI to allow at most 5 users at a time. The size of the screen will also constrain the user interface to display only the relevant data at any given time, offering additional options for other features.

The user interface is constrained to be touch-friendly, i.e. the size of a single interactive element must be greater than the width of an average finger and a mouse must not be necessary to perform any operation.

A professional environment constrains the user interface to be free from distractions and as efficient to use as possible. The nature of the confidential topics likely to be discussed in the system also adds constraints the use of straightforward networking calls limiting the minimum performance overhead.

The current availability of the latest Android API in devices will constrain the Android client to use only features in Android 4.0 (*"Ice-cream Sandwich"*).

A cumulative development time of 60 hours per person limits the possibility of a full featured implementation release.

## 2.7 Assumptions and dependencies

The assumptions made by the development team that the end user will be able to adhere to are as follows:

- usage of an adequately sized (27 inches or greater) touch screen capable of handling 10 simultaneous touches;
- the operating system that the client application will need to run on is Windows 8/8.1 with the latest version of Java installed;
- the individual attendees of the conference are also assumed to own a device with the Android OS with connection to the network; alternatively, the attendees may also own a keyboard for input purposes;
- the server is assumed to be located in a cooled environment with the ability to be online continuously or at the bare minimum, at least during the times conference calls will be taking place;
- the server machine has to be connected to a strong network connection as well as having processing power to allow for complex string/data analysis algorithms to take place.

## 2.8 Acceptance criteria

We have devised **three** prototype models that we expect to build throughout our development (with each building on the previous one):

- 1<sup>st</sup> prototype (codename ' $\delta$ '): Build a stable system that contains one server and one client and displays useful output on the touchscreen;
- 2<sup>nd</sup> prototype (codename ' $\lambda$ '): Implement a smartphone application to serve as a client, and have an initial model for server analytics;
- 3<sup>rd</sup> prototype (codename ' $\xi$ '): Have a fully implemented touchscreen interface with support for touch gestures, while building upon what was previously done.

We will consider successfully reaching the 3<sup>rd</sup> prototype to be the main acceptance criterion for this project, as it is a reasonable expectation for the amount of time given. *Appendix B* (Further Work) describes some of the functionalities that could be implemented in the event that we are able to exceed this requirement, as well as a few additional ideas that further build on that.



# A | Management Strategy

We have unanimously agreed that both technical and non-technical work involved in this project shall be evenly split among the members of the group as much as it is possible, and that mn407 will handle the communication with the client and the Computer Laboratory on behalf of the group. More detailed description of the technical work breakdown will be given in the *project plan*.

Throughout the **specification defining and planning phase** of the project, we have agreed to communicate our ideas in several group sessions and using collaborative tools online, all in the purpose of deciding on a common view of what the core concepts of the project should be. This was a successful collaborative effort that resulted in this document and the project plan.

We intend to devise and write tests for each module before starting the actual **software development phase**, in order to avoid any confusion about the requirements. General progress as well as any problems that arise during the individual module development and testing stage will be addressed on regular scrum meetings.

During the **integration and testing phase** it is our intention to consistently communicate in order to make sure that all the module dependencies have been satisfied, with appropriate inter-modular tests, as well as global tests for all modules combined.

Finally, we will prepare for the **project presentation** by meeting two times to discuss the key concepts, successes and challenges throughout the project and a clear manner in which to present this material.

## B | Further Work

Plenty of further work to enhance functionality of the product is possible which will be implemented, time permitting. The first of them would be to integrate voice messaging resembling phone conversations where each attendee would be wearing a headset associated to the conversation they are currently partaking in. This additional feature naturally requires an improvement in the analytics algorithms used to label each conversation and sound processing would be required. Additional representations to visualise the data involving the conversations can also be incorporated into the system, such as displaying conversation topic trends as a graph. The ability to remotely share and annotate various types of documents such as spreadsheets, PowerPoint presentations and source code. Finally, an iOS client application similar to the Android one should be developed in order to allow greater flexibility for the attendees participating.