

**TeamTalk**

**Connecting Your Workspace**

A Web-Application by **Jad Mrad**

**11/30/2018**

**Supervisor: Mohamad Khalil**

**Summary:**

Team Talk is an office chat application made as per request of IDS as part of the internship program, catered to the needs of any company from small startups to large businesses. It offers a simplistic design coupled with a layered coding schema that makes best use of web development frameworks to give the user a great experience. Team Talk facilitates the communication between colleagues/team members through text messaging and even sharing images. Everything is managed on a database to prevent any data loss and any access to the system is done through a login system for added security.

Equipped with different roles with the possibility of expansion and the ability to host multiple companies simultaneously, the system is very customizable. TT can also be fitted to host multiple teams within the same company with the possibility of having all cross-team communication on a secure, manageable system.

**Table of Content:**

1. **Introduction………………………………………………………(2)**
2. **Requirements Engineering………………………………..(3)**
3. **Design…………………………….………………………………….(4)**
4. **Implementation.....…………………………………………… (5)**
5. **Conclusion………………………………………………………..(6)**
6. **Evolution…………………………………………………………..(6)**
7. **References………………………………………………………..(7)**

**Introduction:**

TeamTalk aims to provide a reliable alternative to time-consuming email conversations and pointless conference calls that still offers security and the ability to share images and documents both online and offline. The application can be used in any office setting with different assignable roles to allow managers more control. TeamTalk relies on simplicity and minimalism in design which allows us to reach out to a broader audience and in turn boost usability.

**Requirements Analysis:**

* Reduce Excessive Email & Waiting Around for Responses
* Eliminate Pointless Meetings & Conference Calls
* Save Time from Co-Worker Interruptions & Chatter
* Send offline messages in case the employee is not online
* Send and receive files, attachments, images… through the chat window
* Security Management (Users)
* Company Management (Managers & Admins)
* Chat Section

**Requirements Specification:**

* Have a more direct way of communication (real-time chat)
* System must run completely virtually with no physical connection required
* Isolation within chats preventing outside interference
* Logging messages for monitoring purposes
* Message database for offline chats
* Allow image/file sharing
* Member accounts with User/Manager/Admin roles
* Company accounts that Managers can edit
* Admin account with access to the entire system
* Notification bar
* Recent chats bar

**Design:**

* **Database:**
  + ASP Membership Provider tables
  + myUser (different than default ASP user table)
    - UserID
    - Name
    - Gender (Nullable)
    - Email
    - StartedWorking (Nullable)
    - Title
    - Password
    - (FK)PhotoRef (Nullable)
    - isOnline
    - lastOnline (NULLABLE)
    - Company
  + Company
    - (PK)CID
    - Name
    - Description
    - Email
    - (FK)LogoRef (Nullable)
    - Active
    - RelatedUser (Nullable)
    - ListEmployees
  + Messages
    - (PK) MID
    - (FK)USenderID
    - (FK)uReceiverID
    - timeSent
    - isRead
    - timeRead
    - isUrgent
    - isDeleted
    - isText
    - Text
    - isFile
    - FileRef
  + FileHolder
    - FID
    - FName
* **Pages:**
  + Login
  + Main Page
  + Manage User
  + Manage Company

**Implementation:**

* Phase 1 (Nov1->12):
  + Database Construction
  + ADO.NET Entity Framework
  + ASP Membership Provider
  + Login Page (login.aspx)
    - Entire page
  + Main Page (index.aspx)
    - Top navigation-bar
    - Bootstrap Shell
  + Design
    - Icons & Images
    - Logo
* Hiatus (Nov12->19): [Exams week]
* Phase 2(Nov19->23):
  + Make messaging page a part of Main Page
  + Main Page
    - Sidebar [JQuerry]
    - Text Messaging [C#]
    - Replaced Default Membership Provider with Custom EF-Based Version [Less redundancy with unused tables in DB]
    - Recently Contacted list
    - Urgent Messages
    - New message icons
  + Manage-User and Company-Management Page
    - Front End
* Phase 3(Nov23->25):
  + Main Page
    - Table of employees [JS]
    - File Sharing [C#]
    - Moved File/Image-saving to a temp folder for prototyping reasons
  + Manage-User and Company-Management Page
    - Form [C#]
    - EF Connection
    - Upload images
  + Admin panel page
* Phase 5(Nov26-27):
  + Split main page into two (contacts and chat) to lower cookie usage and delays
* Phase 6(Nov29):
  + Minor bug fixes & final testing
* Phase 7(Nov 30):
  + Added ability to upload files

**Conclusion:**

I believe that I managed to achieve most of what I had set out for this project. Though the farther I got in the more I wanted to add, I checked off all the requirements based on the timeframe I had. I would consider it a success as the system is currently usable and can function as a perfect base to build on for a more customized chat application.

**Evolution:**

Possible ways to improve on the system:

* Allow manager controlled cross-company connections for partnered companies
* Add multi-user chats
* Real-time message updating
* Allow sending broadcasts
* Optimize file-saving with compression and LAN(company-wide) proxy
* Re-instate ASP Membership Provider with a more in-depth user-relationship scheme to make use of all the provided functionalities
* Add end-to-end encryption for extra security
* Create log system for Managers to monitor chats
* Include file sizes when downloading

**References:**

* Multiple stackoverflow.com posts
* WhatsApp web for design references
* Engineer **Prashanth Pamidi**, **HTech** (http://jsfiddle.net/5HXaQ/5/)
* [**http://www.entityframeworktutorial.net/what-is-entityframework.aspx**](http://www.entityframeworktutorial.net/what-is-entityframework.aspx)
* **https://www.codeproject.com/Articles/281573/ASP-NET-Membership-and-Role-Provider**
* **https://www.c-sharpcorner.com/article/entity-framework-introduction-using-c-sharp-part-one/**