

**CS4065 Web Infrastructure**

**Prof. Conor Ryan**

Group 5

Mary Cronin - 0510661

Adrian O Sullivan- 16230124

Deirdre Shanahan- 16230256

Cornelius Broderick – 9119124

Final Project Report Chance Saloon Dating Website

Contents

[Introduction 3](#_Toc521006920)

[High Level Functionality 3](#_Toc521006921)

[Access Information 4](#_Toc521006922)

[Key Functionality 5](#_Toc521006923)

[SPECIALS/Additional Functionality Added after Design Document Submitted 8](#_Toc521006924)

[Out of Scope: - 8](#_Toc521006925)

[Database 8](#_Toc521006926)

[Sources 9](#_Toc521006927)

# Introduction

Introduce your site and tell us about it, e.g. what sort of people it is aimed at. You should be able to reuse much from your initial design here

Remit for the Web Application was to develop and produce an online dating agency. The scope of the project was to provide the user with basic functionality to allow users to register and describe themselves, as well as browse other users and perform various searches (e.g. list all men over 25 who listed swimming as a hobby).

Please note we have rebranded the site and called it **CHANCE DATING** replacing **FIRST CHANCE DATING**

# High Level Functionality

This will be virtually the same as the design document, but this is where you should note if something hasn't been addressed

This website will support multiple users who are seeking a relationship.

The System will automatch pairs of compatible profiles.

In addition a user has the facility to BROWSE/SEARCH and identify people they would like to Match with.

The system will facilitate the communications between these two users once both users **like** each other.

The system will also support an administrator role whose function is to act as a mediator between users and will have the power to block or bar users.

The system will monitor communications between users and automatically block messages that include inappropriate content.

Users will register their email address along with some personal details when setting up a new account. Once this is complete they will have access to do the following:

* Receive recommendations for people who match their profile
* Complete searches/browse of the database to find people
* Chat with other users
* Maintain account by updating profile, contact details or password
* Report inappropriate behaviour

For our website design we have chosen to take a minimalist approach in the number of questions being asked of the user. Based on our research we have concluded that physical attractiveness plays a major part in initial encounters (Swami & Furnham, 2008). The website will be limited geographically to Ireland

# Access Information

Give us the landing page on hive for the website along with any passwords (for example your group's database password on hive, or what your site's administrator password is, if there is one).

**Please use GOOGLE CHROME or FIREFIX**

**Links to Pages**

|  |  |
| --- | --- |
| Dating Pages | Page Link |
| Index.php | <http://hive.csis.ul.ie/4065/group05/index.php> |
| Logon.php | http://hive.csis.ul.ie/4065/group05/Logon.php |

**Useful Passwords**

|  |  |  |
| --- | --- | --- |
| USER | EMAIL | PASSWORD |
| SysAdmin | [16230256@Studentmail.ul.ie](mailto:16230256@Studentmail.ul.ie) | welcomE01# |
| SysAdmin | 16230124@studentmail.ul.ie | welcomE01# |
| SysAdmin | 0510661@stedentmail.ul.ie | welcomE01# |
| SysAdmin | 0199124@studentmail.ul.ie | welcomE01# |

# Key Functionality

Address each of the key functionality points from the project description to tell us exactly what your project does. Current list – Deirdre to update with latest version

# SPECIALS/Additional Functionality Added after Design Document Submitted

Geo Location Matching – we calculate the distance between users using Latitude and Longitude coordinates and return matches for people within their specified travel distance range

Chat Facility – Includes moderation of black listed words and build in facility for suspending accounts for repeat offenders

Chat Facility - Support the concept of Mutual Consent. If both users do not like each other then they cannot initiate conversations

SQL Injection Prevention - All database queries where users provide input are processed using parameterized queries

PHP Injection Prevention – All Form submission completed using HTML special characters

Session Management – Each logon is assigned an encrypted session hash which is stored on the user profile in the database. This session hash is checked on each page to ensure user account is not been spoofed

Syst Admin - In addition to agreed scope to support an administrator role described above. We have introduced the functionality to allow the SysAdmin to revoke a suspension or revoke a BAR of a user.

The website is responsive allowing the user to use the app on mobile or other screen of different sizes.

# Out of Scope: -

Email Functionality – As agreed with Aidan and Connor the app will provide the user the option to reset their password and enter email address to send password. The email functionality is not in scope for this app.

# Database

List your database tables. If they haven't changed from the initial design then it is fine to simply copy and paste from the earlier submission

The following changes has occurred in the database tables from the initial design:

* user\_profile --> 2 new records added
  + suspended\_until\_date datetime added
  + session\_hash varchar(250) added
* user\_communication --> one record removed
  + Status\_date removed
* status (table name changed from status to status\_master)

The changes are highlighted below.

Also new view created:

1. matches\_view

Also cursor created:

1. Cursor to get all un-processed engine usage entries for period

## Database Tables

## user\_profile

Table 1 user\_profile, This table describes each user. The primary key is id, each user has email as unique identifier, with user\_status\_id as foreign key which links a central status table. The password\_has is secured using sha2 encryption with a 256 length key.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Primary** | **Foreign** |
| id | int | Yes |  |
| password\_hash | Varchar(200) |  |  |
| first\_name | Varchar(50) |  |  |
| surname | Varchar(100) |  |  |
| email | Varchar(100) | (unique) |  |
| date\_of\_birth | date |  |  |
| gender\_id | int |  | Yes |
| gender\_preference\_id | int |  | Yes |
| From\_age | int |  |  |
| To\_age | int |  |  |
| City\_id | int |  | Yes |
| Travel\_distance | int |  |  |
| Relationship\_type\_id | Int |  | Yes |
| picture | blob |  |  |
| my\_bio | Varchar(1000) |  |  |
| Black\_listed\_user | Tiny int |  |  |
| Black\_listed\_reason | Varchar(100) |  |  |
| Black\_listed\_date | date |  |  |
| User\_status\_id | int |  | yes |
| is\_administrator | boolean |  |  |
| suspended\_until\_date | datetime |  |  |
| session\_hash | varchar(250) |  |  |

## match\_table

Table 2 match\_table, This table lists the matches for a user, Primary Key for each match is id, with foreign keys( match\_user\_id\_1, match\_user\_id\_2, match\_status\_id) linking users and foreign key - communication\_id identifying if there is an initial communication between matches.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Primary** | **Foreign Key** |
| id | int | yes |  |
| match\_user\_id\_1 | int | (composite unique) | yes |
| match\_user\_id\_2 | int | (composite unique) | yes |
| match\_date | datetime |  |  |
| response\_date | datetime |  |  |
| user\_id\_1\_interest\_level | int |  |  |
| user\_id\_2\_interest\_level | int |  |  |
| communication\_id | int |  | yes |
| User\_1\_match\_status\_id | Int |  | yes |
| User\_1\_match\_status\_date | Datetime |  |  |
| User\_2\_match\_status\_id | int |  | yes |
| User\_1\_match\_status\_date | Datetime |  |  |
| system\_generated\_match | boolean |  |  |

## user\_communication

Table 3 user\_communication, This table is the communications table between user, Primary Key for each communication is id, with foreign keys( from\_user\_id, to\_user\_id status\_id) linking users and foreign key - black\_listed\_word\_id identifying inappropriate communications made by users. Black listed communications will be blocked and if a predefined quota is reached by a user then their account will be suspended. When a reply to a communication is recorded, a link is established to the previous communication in order to allow the conversation thread to be maintained.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Primary** | **foreign** |
| id | int | Yes |  |
| from\_user\_id | int |  | yes |
| communication\_datetime | datetime |  |  |
| message | Varchar(140) |  |  |
| status\_id | int |  | yes |
| to\_user\_id | int |  | yes |
| replying\_to\_communication\_id | int |  | yes |
| black\_listed | boolean |  |  |
| Black\_listed\_date | datetime |  |  |
| Black\_listed\_word\_id | int |  | yes |

## black\_list\_word

Table 4 black\_list\_words, This Table contains a list of black listed words, it is used to check if the user uses inappropriate language and block any communication where one of the words are identified. Primary Key is id which is unique to each black listed word. If a user reaches a predefined quota of inappropriate communications then their account will be suspended. In order to prevent offence to people who have access to the database table, we have encrypted the word using md5 encryption, while this is no longer considered the securest form of encryption, it serves a purpose in this instance.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Primary** |
| id | int | yes |
| word | Varchar(100) | unique |

## user\_interests

Table 5 user\_interests, This table contains the lists of interests a user has. Primary key is user\_interest\_id.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Primary** | **Foreign** |
| Id | Int | yes |  |
| Interest\_id | Int | Unique (composite) | Yes |
| User\_id | Int | Unique (composite) | Yes |

## interests

Table 6 interests, This Table contains a master list of interests a user may have, it is used by the match table and user interests table. Primary Key is interest\_id which is unique to each interest

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Primary** |
| Interest\_id | int | yes |
| description | Varchar(200) |  |

## Gender

Table 7 Gender Table, This is used by the user Profile table to identify the gender and gender preference of the User. Primary Key - id ,is an identifier for a gender type.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Primary** |
| id | int | yes |
| Gender\_name | Varchar(200) |  |

## city

Table 8, The city Table, is unique to each user and identifies the location and location preference of match, Primary key is User\_id. Geo-coordinates are recorded for each city so that distance between user locations can be calculated.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Primary** |
| id | int | yes |
| City | Varchar(100) | unique |
| county | Varchar(100) |  |
| geo\_x | float |  |
| geo\_y | float |  |

## relationship\_type

Table 9 Relationship Type, is used by the user Profile table to identify the relationship type the user seeks. Primary Key - id, is an identifier for a relationship type.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Primary** |
| id | int | yes |
| Relationship\_type | Varchar(200) | unique |

## Status\_master

Table 10 Status\_master Table, Primary Key - id, is an identifier for user status. The status table stores status for the user profile, match table and user communications. These are separated using a Boolean indicating which status relates to which table.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Primary** |
| id | int | yes |
| Status\_description | Varchar(100) |  |
| is\_user\_status | boolean |  |
| is\_match\_table\_status | boolean |  |
| is\_user\_communication\_status | boolean |  |

# Stored Procedures

## generate\_matches

This stored procedure can be called with a from and to user profile id to support processing either a single user profile or a range of profiles. When executed it carries out the following steps:

* Get a list of new or active user profiles and for each one do the following
  + Identify a list of matches based on their preferences who are within the target distance of their city
  + Exclude profiles that they have already been matched against
  + Insert any new matches into the match table, these will be flagged as auto generated.

When both users log onto the system they will be able to browse their list of matches and the auto generated matches will be listed. Each user can status a match with options such as Like, Maybe or Goodbye. If status of Goodbye is selected then this match will never be presented to the user again.

# Sources

List all images that you have taken from other sources

Photos taken from <https://www.pexels.com/>

All Icons are sourced from Open Source Project