Software Design Specification: DTP

**Draft 1**

*March 10, 2021*

**Group 2**

Michael Liguori and Christopher Chen

**Contents**

[**1 Introduction**](#_gjdgxs) **1**

[1.1 System Overview](#_ackzeor3r5ny) 1

[1.2](#_ackzeor3r5ny) [Design Map](#_lyzx5bxceu0m) 1

[1.3](#_ackzeor3r5ny) [Supporting Materials](#_ij8jp3bs69rk) 2

[1.4 Definitions and Acronyms](#_lcinks8ozlq6) 2

[**2**](#_gjdgxs) [**Design Considerations**](#_thm4ltoaot0i) **4**

[2.1 Assumptions](#_prqtikd1nlla) 5

[2.2](#_prqtikd1nlla) [Constraints](#_jjoeo8vagfa1) 5

[2.3](#_prqtikd1nlla) [System Environment](#_qkfeluhbya7c) 5

[**3 Architecture**](#_ih5uvrwdgd60) **6**

[3.1 Overview](#_or95r4cpvgqv) 7

3.2 [Rationale](#_nvrst2s2zh3q) 7

3.3 [Component Details](#_iqv8vudgwe5) 8

[**4 High Level Design**](#_l9fnecjsqjix) **15**

[4.1 Conceptual View](#_8roi37mts3ht) 16

[**5 Low Level Design**](#_xltxu9ukibel) **18**

5[.1](#_8roi37mts3ht) [Module 1..n](#_2offh17ohy6g) 19

[**6 User Interface Design**](#_tkm0ivlkz6lz) **20**

6[.1](#_8roi37mts3ht) [Application Control](#_sx1onk8zwu8) 20

6[.2](#_8roi37mts3ht) [Screen 1..n](#_3hcg3yoroe26) 20

# 1 Introduction

## System Overview

This application is a platform for gamers to quickly host, queue up, and create rooms for play for multiple games for multiple platforms at once. The application will allow registered users to select several games they are interested in playing by either queuing up for or hosting a queue for their desired games. By achieving this, the app hopes to connect gamers who are ready to play quickly and with more efficiency, give gamers a work around to games’ built-in matchmaking algorithms, to increase longevity of a game’s online presence, and to create a resurgence of online presence for older games without the need of forum searching or posting.

## Design Map

This document contains information regarding design considerations, including assumptions, constraints, methodologies and risks involved. It also includes information on architecture of the system, rationales surrounding the way that the architecture was designed, and a detailed roster of the components that the system is composed of. It will also have a high-level conceptual view of the system, explaining components of the system with diagrams and basic high-level descriptions of these units.

## Supporting Materials

This application is intended to be initially developed on a PC with three main relationships which are the Client, Web Server, and Database Server. The Database and Web Server will be developed on and for Windows OS. The Client will request and receive services from the Web Server through the use of any major web browsers such as Google Chrome, Apple Safari, and Mozilla Firefox.

References to any system requirements can be found in the following links:

* Windows OS –https://docs.microsoft.com/en-us/windows/
* Google Chrome –https://developer.chrome.com/docs/
* Apple Safari –https://developer.apple.com/safari/resources/
* Mozilla Firefox -https://developer.mozilla.org/en-US/docs/Mozilla/Developer\_guide

## 1.4 Definitions and Acronyms

* Administrator User: A user of our system who oversees other users’ accounts and upkeeps the integrity and security of all databases the system uses.
* Application Programming Interface (API): An interface used in computing which receives requests and sends responses to a client user.
* Client: A PC or program which accesses services available from a server. In this application’s case, a web server.
* User: A User of our system who is only allowed to use basic, non-paid functions of the application.
* Database: A data storage structure that stores information in an organized matter.
* Database Server: A server which provides database services to other computers or programs through the use of a database application.
* End-user license agreement (EULA): A legal contract entered into between a software developer or vendor and the user of the software.
* Graphical User Interface (GUI): An interface which is used by users via visual components that can be interacted with, these components conveying information visually.
* Operating System (OS): A piece of system-based software that manages both hardware and software for a computing device.
* Personal Computer (PC)- a computer which is used by one individual at a time
* Programming Language: A language that creates output based on an instruction set coded into it.
* SQL: A programming language which is used in the creation and maintenance of databases.
* User Interface (UI): Is the point of human-computer interaction and communication in a device
* Unified Modeling Language (UML): Is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.
* Web Browser – a software application for accessing data on the Internet
* Web Server- A computer that stores web server software and a website’s files and components

# Design Considerations

## Assumptions

We assume that third-party components that we plan to use with the application are fully functioning with minimal errors. These components include:

Dependencies: Node js, Web Browsers, MySQL Workbench

Dependencies on Node packages: Express js, Pug js, Socket.io, MySQL

## Constraints

* Ability to keep internal application database updated from the API efficiently
* Security considerations with the use of two-key authorization.
* Automatic server timeout if the user hasn’t interacted with the app for a certain amount of time (360 seconds)

## System Environment

This application will work with a PC using Windows OS for the Web and Database Servers and any other OS running a supported web browser for the Client. The software that the server will operate in is Windows and for the client will depend on its operating system that runs the web browser. The hardware this application will use are any pc running windows for the server capable of handling the storage of data and the sending and receiving of data from clients to other clients. The application on the Client must also interact with the internet through a WiFi connection or an ethernet connection to a network in order to access the Web Server. The Web Server and Database Server will be on the same PC.

# Architecture

## Overview

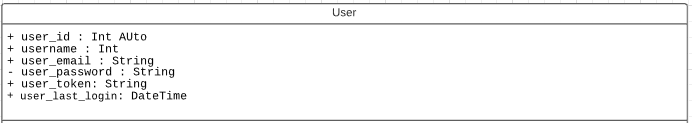
DTP consists of the application located on users clients pcs, as well as a web server and database server which holds both user data and information that is to be sent to the user when requested (data on games, user data, etc.). The application consists of components including a queue up feature, game search, user profile editor, add friends, set preferences, view friend profile, and view friend list. The aforementioned information for these components is stored server-side in a secure database which only administrative users of the server are able to access and change. The application runs on the user’s PC and is connected to the DTP Web server which retrieves data from the Database Server for use in the main application when needed.

## Rationale

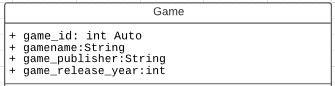
The server-client architecture was chosen to conform to the 2.3 design standards as well as to reinforce the security and legitimacy of data being distributed to all users of the application. These security standards and permissions are important when executing CRUD commands and interfacing with the database. The web server side that provides the data to clients from the database is only admissible by administrative users who will always oversee the legitimacy of the data being distributed application wide. This rationale will reinforce the quality of data within the app improving the user-end product.

## Component Details

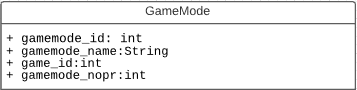
This Section of the document will showcase the UMLs that will eventually be designed once a final version of the application is complete. For now, the few entities we have the in Database are User, Game, GameMode, QueuedPlayer.



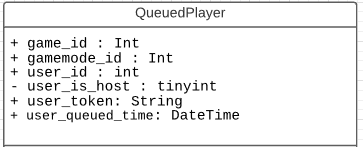
This is the user class that represents the columns and datatypes that are associated with a user’s account thus far. An arbitrary user\_id is created for each new user, and their username, email, and password are stored. The token represents a randomly generated string which is used to identify a user throughout different web pages when logged in to DTP. The last login is updated whenever the authenticated user requests any activity to the DTP web server, and is checked to see whether or not they should be logged off for inactivity



This is the game class that represents the columns and datatypes associated with a game’s information in the database thus far. There is an arbitrary number associated with a game entry and holds information to identify the game such as name, publisher, etc. This is the parent class of the gamemode class.



This is the gamemode class that represents the columns and datatypes for the gamemode information for a particular game. It is identified with the foreign key game\_id from the game table. It has an arbitrary number associated with its name and game and has the number of players, nop, for each gamemode.



This is the queuedplayer class which represents the information that is stored in the database when someone queues for a game/games. Each game queued will store the game\_id, gamemode\_id, the user\_id, whether the user queues as a host or not, and the time they began their queue. This will allow the server to make checks against the database to find hosts and players to fill the host’s queues.

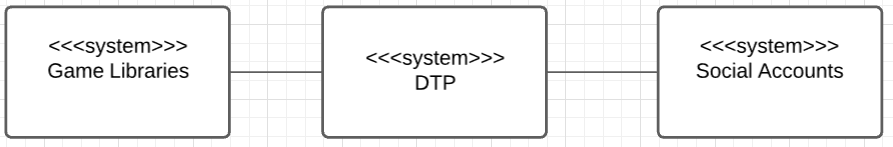
The final code will be organized into different classes for the database sql query functions, the index js file for node functions, the express js function and other possible classes that may be created to organize the code.

# High Level Design

## Conceptual View

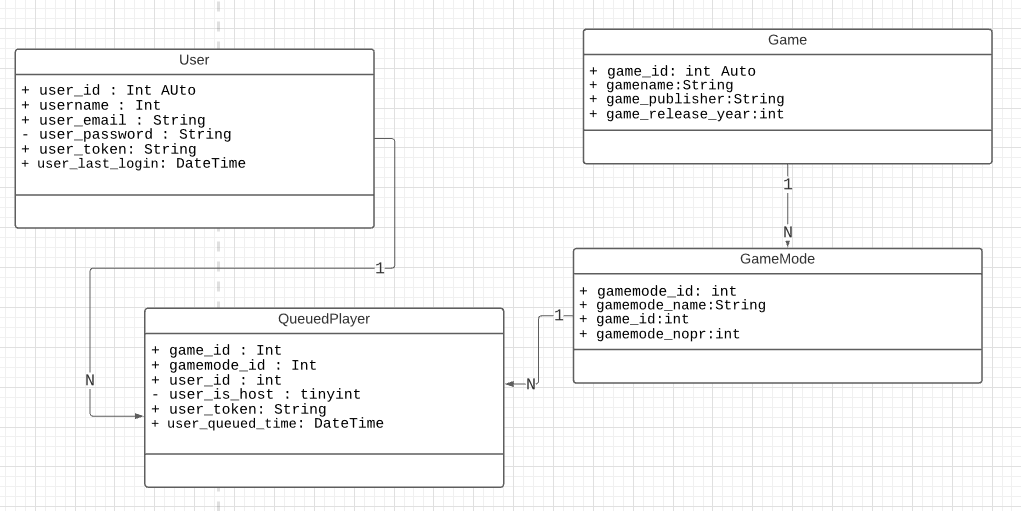
Eventually, our system, in addition to its own features, will interface to external instruments to function completely in its own environment. These external instruments include APIs potentially to other game libraries and databases and external accounts, to better facilitate rooms for our users. For example, being able to connect DTP user accounts with third-party accounts such as steam, google, Facebook etc.

The reason for using the external APIs is simple. To import game libraries by a user’s account allows them to see all the games they own for that account in the DTP application. Social media APIs will help users find friends outside of the gaming world if they so choose.



# Low Level Design

## Module 1..n



The User Class will queue for game modes of games.This will place them into the queued player class. A User can be queued for many game modes.

A game mode is a subclass of the game Class. A gamemode can be queued by many players.

# User Interface Design

## Application Control

The common look within the app is a minimalistic design with a straightforward graphical user interface. The color scheme of the app will consist of tones of professional greys with luxurious purples, creating an easy to follow and comforting experience for users. Common behavior between all screens will include a side menu which can be tapped to manually navigate to a specific screen, denoted by three horizontal lines, one on top of another (colloquially known as the “hamburger button”). Most menus will be in the form of lists, where each menu item is a button on a vertical stack of options, each leading to the screen indicated by the words on the aforementioned button. Title bars will in most instances contain the time as well as the name of the application (“DTP”), while some may only contain the latter. Being a web application, most interactions with the application screens will be in the form of html and JavaScript events interacted with by the user through mouse and keyboard.

## Pages

Page 1: Sign In

On this page, the user will be provided a login option which includes two textboxes: one for a username, and one for a password. Then the user will click the sign in button to sign in. If the user has not registered yet, a create an account button is provided at the bottom of the web page.

Page 2: Registration

On this specific page, the user will be provided an option to sign up: this web page includes four text boxes for the desired username, the password, the email, and the confirmation of the password entered. The user will then press the “create an account” button to register for the application. If they have already signed up, the user can redirect back to the Sign In page with a link at the bottom of the screen.

Page 3: Home

On this page, the user will be presented with a menu with various buttons to access most application features, including “My Queues,” “My Account”, “My Friends, and “Games”

Page 4: My Queues

On this page, the user will be presented with a list of their current queues, as well as other information on current users in queues and their messages. Additionally, the user will have the option to cancel the queues here.

Page 5: My Account

On this page, the user will be presented with options to change their profile and account details, such as username, password, profile picture etc.

Page 6: My Friends

On this page, the user will be presented with a search text box to search for users, as well as a list of current friends and options to remove or direct message them.

Page 7: Games

On this page, the user will be able to view a list of games and queue for which ones they are down to play.