

# CS 341 Project (Project Themis)

**Created by:**

Byron Lewandowski (lewandowskib@etown.edu)

Devin Drennen (drennend@etown.edu)

Matt Kowalski (kowalskim@etown.edu)

Alpha Sow (sowa@etown.edu)

Alex Good (gooda@etown.edu)

<https://www.sharelatex.com/project/58afb4f045b9839a55ff357f>

<https://github.com/DevinDrennen/Project-Themis>

<https://www.dropbox.com/home>

**March 1st, 2017**

## Contents

<b>1</b>	<b>Customer Statement of Requirements</b>	<b>5</b>
1.1	Introduction . . . . .	5
1.2	Background . . . . .	5
1.3	Devices and Specification . . . . .	6
<b>2</b>	<b>Glossary of Terms</b>	<b>7</b>
<b>3</b>	<b>System Requirements</b>	<b>8</b>
3.1	Functional Requirements . . . . .	8
3.2	Non-Functional Requirements . . . . .	8
3.3	On-Screen Appearance Requirements . . . . .	9

## List of Figures

## List of Tables

### 3.3 - On-Screen Appearance Requirements

# 1 Customer Statement of Requirements

## 1.1 Introduction

Our Project will be about creating a collection of games with help of two clients: Dr. Gu and our other classmate's teams. For our intended system, which can be referred as a collection of games, will need the following: **clients, resources, equipment, and our team members.** For clients, we have both our professor and the other team groups. Our intended game system must be played as human versus human. There will be no human versus computer interaction because of the complexity of the system. Next, resources will be a 5 student-team (diverse set of skills) and the time to finish the proposed game system. Another beneficial resource will be the help and guidance from our professor. The equipment needed will be Java (Eclipse), IDE's, computer(s), MySQL, and team members. All of these resources are essential to the project and are needed to get the system functional. Lastly, our team members will be needed for our system because without the diverse set of skills of a 5 person team experts in java, databases, and/or other skills.

Other systems such as the gaming websites Miniclip, Pogo, Y8, and others use a collection of various genres and types also know as a gaming collection. They are much more advanced systems, but still use the common team, resources, and equipment so the system can operate correctly. Our team is motivated to create a collection of games which will allow clients to freely entertain and continue conversation and interaction amongst people. Starting off we do not have any previous version history, our current version is Project Themis Version 1.0.

## 1.2 Background

Since we are doing a collection of games, there will be minor similarities of other game systems such as websites like Miniclip, Pogo, Y8, and others. The main difference is we are doing a currently working on one finished full game. After the the one game is finished, we will look into our list for other games in mind to develop and create. In the early stages of our project we created user stories for some games such as Battleship, Checkers, Chess, and Othello. In comparison to other gaming systems, Battleship-game.org allows you to play the game online and/or with a friend. 247.7 Checkers lets you play either by one player and two. Chess.com allows you to play chess against the computer and or human player. Lastly Othello-on-line.org has a version of Othello that lets you verse the computer and play against an opponent as well. Our most unique feature would be just having the 1 on 1 game play against a real person, also having a database which would store our game's data in the server we create which can be accessed without the

server being physically there. The feature of simplicity and accuracy will be a positive pointer for our gaming system collection.

### **1.3 Devices and Specification**

Requires Windows, Mac, and or Linux operating systems. The language used for Project Themis version 1.0 is English. Code will be written in Java and MySQL.

Our game collection system will only be available on computers, no phones or tablets.

## 2 Glossary of Terms

**Database** - Databases are a place to store information. In our case the database will store the collection of our games so they can be accessed.

**Clients** - Clients are the specific people whom will be using our game collection system, in this case we have two clients (Dr. Gu and the Other teams).

**Java** - Coding language used to run on an IDE which can be used to create client-server web applications. For our case we are using the code to create the games in the Java language.

**Computer** - An electronic device for storing and processing data. In our case we are using it to run and create our game collection system. Very important we have more than computer

**LaTeX** - a document preparation system which uses plain text to the opposed formatted text. We are using this to create a PDF of our project pre/final report.

**API** - Also known as application program interface is a set of routines, protocols and tools for building software applications. In this case we are using different API's to create the different applications of our intended game collection

**SQL** - Query language used for requesting information from a database. We are using it to store our game collection in it then be able to access it through the queries.

**De-bugging** - Also known as debug, is to identify and remove errors from a computer hardware and or software. In our case we would test and check our code in Eclipse, LaTeX and SQL/MySQL.

**Slack** - Brings all communication together in one place for a team. We are using that as our team communication for this project.

**Dropbox** - a file hosting service which allows for file storage, synchronization, and other personal cloud integration. We are storing our overall project information/code in dropbox.

**GitHub** - a web-based version control repository and internet hosting service. We are using this to distribute our version control and source code within our repository/group GitHub

### **3 System Requirements**

Below are the basic functional and non-functional requirements for our system:

#### **3.1 Functional Requirements**

The user should be able to start our game system by running the .jar file. Next they should be prompted to log-in to the system using their log on user name and password. From there a menu for the system should appear and the user should be able to click on a game of their choice and it should start a new game. Once they have made a move the system should send the game data to the server where it will be stored for the second player to access. Once the second player accesses the game they make their moves. A possible option for our system will be to add a help menu that when clicked on will explain the basic moves of the game being played.

#### **3.2 Non-Functional Requirements**

The system should have a relatively quick response time when it comes to user input. As far as response time from the server it should not take more than a few minutes for the system to send the game data to the server and the server to update the game for the second player. The system should be available at any time of the day. As far as reliability the system should not be down for more than a couple hours unless it is a server hardware issue. We should be able to maintain the software end of the system at any time making maintenance relatively quick and easy. The server will be secure by using a hashing function to store the users log-on information.



### 3.3 On-Screen Appearance Requirements

Identifier $q$	Priority	Description
REQ-1	3	Requirements must include a feasible size playing interface with readable icons, buttons, and fonts.
REQ-2	2	A possible on screen help button, which may list the possible controls for each game. For example mouse clicks, and or keyboard specific strokes.
REQ-3	5	The User interface <b>shall</b> have user friendly icons and other simplicity options to help show the user where to start, stop, pause, and expand the game.
REQ-4	5	The Game-Collection System <b>should</b> be user friendly and simple to use.