

Timekeeper/Statistical tool for Basketball  
Sponsor: Prof. Wayne Dyksen & MSU Basketball Team  
Spring 2004

## BID DOCUMENT



Team 2  
Edward Bangs  
Bryan Berry  
Chris Damour  
Kim Monteith  
Jonathan Szostak

## Table of Contents

1.0: Introduction .....	3 - 7
1.1: Statistical GUI .....	3 - 4
1.2: Time Management GUI .....	5 - 6
1.3: Bench Viewing GUI .....	7
2.0: System Summary .....	8
3.0: Spiral Model .....	10
4.0: Known Data Models .....	12

## Introduction

The goal for this document is a first description of the project, with a level of detail somewhere between a full requirements document and a simple English-language description.

Our actual client is the Michigan State University Basketball Team, but our main contact is Dr. Wayne Dyksen of MSU's computer science department. Dr. Dyksen provides us with requirements of the project.

The MSU Basketball Team would like to upgrade their current statistical and clock management tools from pad and paper to programs designed to work on tablet PCs. The project is broken up into three unique GUI's: the statistical tool, the player tracking tool, and the bench viewing tool. (Possibly a play statistical tool and some others if time permits)

The game statistical tool consists of many parts (it will be used during the game and could possibly be used while watching film). The main goal of this tool is to record all the shooting statistics of only MSU's team during the game. It records free throw attempts/makes, two point attempts/makes, three point attempts/makes, and fouls for each player (separating the statistics by period). It is the user's job to physically record these facts through the GUI buttons. From this information the GUI will also display the shooting percentages of each player and the team as a whole, the total points of each player and the team as a whole, and it will display the team fouls. The tool will incorporate various color visuals to warn the user of a player close to fouling out and also when the team is in the double/single bonus. The user also has the option of highlighting a specific player or statistical row for better contrast and controls what players are viewed along with the arrangement of these players. Each team will be highlighted in their team colors for easy viewing as well.

The GUI will allow for some user error. For example, if the user accidentally records a missed free throw as a made one he can easily correct this through the '-' button near the top.

Other aspects of the GUI will be error proof. For example, the user will not be allowed to give a player more than five fouls.

A prototype screen shot is on the next page.

Player Names and Numbers

Shooting Statistic Buttons

Select Team		20	0/0	0%	0/0	0%	0/0	0%	0	0	0	0	Exit				
LR	SR	R	-	PF	O1	X1	%O1	O2	X2	%O2	O3	X3	%O3	1Pt	2Pt	3Pt	Total
Select Player				0	0/0		0%	0/0		0%	0/0		0%	0	0	0	0
<input checked="" type="checkbox"/> 1	R			PF	O1	X1	%O1	O2	X2	%O2	O3	X3	%O3	1Pt	2Pt	3Pt	Total
Select Player				0	0/0		0%	0/0		0%	0/0		0%	0	0	0	0
<input checked="" type="checkbox"/> 2	R			PF	O1	X1	%O1	O2	X2	%O2	O3	X3	%O3	1Pt	2Pt	3Pt	Total
Select Player				0	0/0		0%	0/0		0%	0/0		0%	0	0	0	0
<input checked="" type="checkbox"/> 3	R			PF	O1	X1	%O1	O2	X2	%O2	O3	X3	%O3	1Pt	2Pt	3Pt	Total
Select Player				0	0/0		0%	0/0		0%	0/0		0%	0	0	0	0
<input checked="" type="checkbox"/> 4	R			PF	O1	X1	%O1	O2	X2	%O2	O3	X3	%O3	1Pt	2Pt	3Pt	Total
Select Player				0	0/0		0%	0/0		0%	0/0		0%	0	0	0	0
<input checked="" type="checkbox"/> 5	R			PF	O1	X1	%O1	O2	X2	%O2	O3	X3	%O3	1Pt	2Pt	3Pt	Total
Select Player				0	0/0		0%	0/0		0%	0/0		0%	0	0	0	0
<input checked="" type="checkbox"/> 6	R			PF	O1	X1	%O1	O2	X2	%O2	O3	X3	%O3	1Pt	2Pt	3Pt	Total
Select Player				0	0/0		0%	0/0		0%	0/0		0%	0	0	0	0
<input checked="" type="checkbox"/> 7	R			PF	O1	X1	%O1	O2	X2	%O2	O3	X3	%O3	1Pt	2Pt	3Pt	Total
Select Player				0	0/0		0%	0/0		0%	0/0		0%	0	0	0	0
<input checked="" type="checkbox"/> 8	R			PF	O1	X1	%O1	O2	X2	%O2	O3	X3	%O3	1Pt	2Pt	3Pt	Total
Select Player				0	0/0		0%	0/0		0%	0/0		0%	0	0	0	0
<input checked="" type="checkbox"/> 9	R			PF	O1	X1	%O1	O2	X2	%O2	O3	X3	%O3	1Pt	2Pt	3Pt	Total
Select Player				0	0/0		0%	0/0		0%	0/0		0%	0	0	0	0
<input checked="" type="checkbox"/> 10	R			PF	O1	X1	%O1	O2	X2	%O2	O3	X3	%O3	1Pt	2Pt	3Pt	Total
Select Player				0	0/0		0%	0/0		0%	0/0		0%	0	0	0	0
<input checked="" type="checkbox"/> 11	R			PF	O1	X1	%O1	O2	X2	%O2	O3	X3	%O3	1Pt	2Pt	3Pt	Total
Select Player				0	0/0		0%	0/0		0%	0/0		0%	0	0	0	0
<input checked="" type="checkbox"/> 12	R			PF	O1	X1	%O1	O2	X2	%O2	O3	X3	%O3	1Pt	2Pt	3Pt	Total

Foul Statistic Button

The player tracking tool is not as visually complex as the statistical tool above, however there is an underlying complexity due to the need for real time error handling with synchronization of a clock. It will be strictly used for inputting data for the game.

The main goal of this tool is to help MSU coaches keep track of how much a player has played or rested at any given time. The only two responsibilities of the user are to check players in/out during the game and also start/stop the clock. The GUI will visually warn the user when a player has played over four minutes by highlighting the player's name in red (these minutes are in game minutes). When the player is checked out the resting clock will begin and will eventually turn green when the player has rested three minutes (these minutes are in real time, not game time). Behind the GUI the database will also store all the player minute information for analyzing after the game.

The GUI will allow for some user error. For example, if the user forgets to start the clock he will be able to drift the clock forward or backwards in order to sync with the real game clock. Other aspects of the GUI will be error proof. For example, the GUI will not allow the user to check in more than five players or start the clock if five players aren't checked in.

Other aspects of the GUI will be very user friendly for quick use during the fast paced game. For example, the user will be able to highlight current players in the game (while the clock is running) and when the clock is stopped the tool will automatically check them out. As in the previous tool the user has control over what players are viewed along with the arrangement of these players (the order they are listed).

A prototype is on the next page.

Player's Rest Time (Red means not enough rest)

Player's Current Active Time

Drift Clock Backwards

Drift Clock Forward

Spartan Basketball Stats - [Player Timer]

File Edit Insert Records Window Help

Period: 1 Michigan State Spartans Men's Basketball Time: 18:23

↑ ↓

**Clock is Stopped**

Warnings	Activate All	Select Player	Checked Out			Check Player	Checked In				
			Time	Player	Time		Player	Time	Player		
			Current	Remaining	#	Name	In / Out	#	Name	Current	Remaining
<input checked="" type="checkbox"/>		1	0:00	3:00			←	13	Ager	1:37	2:24
<input checked="" type="checkbox"/>		2	0:00	3:00			←	23	Torbert	1:37	2:24
<input checked="" type="checkbox"/>		3	0:03	2:57	15	Anderson	→			1:37	2:24
<input checked="" type="checkbox"/>		4	0:08	2:52			←	3	Brown	1:37	2:24
<input checked="" type="checkbox"/>		5	0:00	3:00			←	40	Davis	1:37	2:24
<input checked="" type="checkbox"/>		6	0:00	3:00			←	5	Hill	0:00	4:00
<input checked="" type="checkbox"/>		7	0:00	0:00						0:00	0:00
<input checked="" type="checkbox"/>		8	0:00	0:00						0:00	0:00
<input checked="" type="checkbox"/>		9	0:00	0:00						0:00	0:00
<input checked="" type="checkbox"/>		10	0:00	0:00						0:00	0:00
<input checked="" type="checkbox"/>		11	0:00	3:00	10	Deacon	→			0:00	4:00
<input checked="" type="checkbox"/>		12	0:00	0:00						0:00	0:00

**Clock is Stopped**

View Game Stats [Info] Check Out All Begin the Period End the Period

Load Roster Open Exit

Start Stop Clock

The bench viewing GUI will be strictly used for viewing and will not be able to change any records in the database (it will only be used during the game).

The main goal of this tool is to keep the coaches updated on the game statistics and playing time of all the players. One of the coaches on the bench will actually have a tablet PC and will be viewing this GUI throughout the game. There will be different views (if time permits, otherwise just one view) depending on what info the coach wants at that moment. All of the info represented by the different views will be pulled from a database that is created by the above GUI's.

A prototype is not available for this GUI at this time.

# System Summary

## Functionality:

The following list defines what must be done to meet the project requirements and what will be done if time permits.

Must be done:

- *Statistical GUI*  
Statistical GUI prototype must be implemented in C# (the current code is in Visual Basic). Some new functionality will be put in (missed two pointers and missed three pointers will also be recorded with this tool). Some changes to other buttons may occur for better screen real-estate and a more user friendly tool. Another change is only keeping stats for MSU, not away teams.
- *Player Tracking GUI*  
Time Management GUI prototype must be implemented in C# (the current code is in Visual Basic). The current GUI seems to work well so we don't foresee any changes in functionality with this tool.
- *Bench Viewing GUI*  
Bench Viewing GUI must be implemented from scratch in C# (there currently is no prototype available for this GUI).
- *Network Architecture*  
The group will decide upon a reliable way to network the tools together in order to create the bench viewing GUI. Currently we will rely on a wireless ad-hoc network to satisfy the physical layer, and Microsoft web services to implement the communications between the machines.
- *Database Structure*  
The group must design and implement the database which will hold all the information from the GUI's. The group must decide on the tables & columns used to store all the specific information. Also a replication scheme must be created in case of system crash.
- *Database Control*  
This is a real time database, limited by the physical capabilities of the tablet machines (Power, interface speed, & processor speed). Therefore a process must be designed in which all the interfaces make a minimal amount of SQL statements to the database, yet still provide all the required functionality. Transactions between the active memory on the tablets and the database storage must be made at optimal times decided by the group. Additionally there is the problem with one machine making a change to the database, and the other machines becoming aware of that change almost instantaneously.



Might be done:

- *Play Tracking GUI*

The Play Tracking GUI must be implemented from scratch in C# (there currently is no prototype available for this GUI). This will be a tool similar to the statistical GUI above but will have functionality for rating and recording a play timeline of the game.

**Reliability:**

The system will be able to recover data where the game left off, in the event of a failure. When the system is back up running, it will appear to the user that the game was stopped. This is a goal that will be implemented if time permits.

**Usability:**

The user can effortlessly add and modify data through user-friendly buttons. The user will be given visualization options of highlighting rows and columns and arranging the list of players for better viewing.

**Design Constraints:**

The various GUI tools must run in SQL Server on a tablet PC. There is a display size constraint due to the small screen real estate of the touch panel. All code will be implemented in C#.

**User Documentation:**

All documentation can be found on our team website <http://wonderwoman.cse.msu.edu>. The format will be PDF. Comments and concerns will be addressed via the website's web-log.

**Software Interfaces:**

Development will be in Microsoft Visual Studio .Net and will use SQL Server.

**Legal, Copyright and Other Notices:**

All plays and basketball information are property of MSU's basketball team.

## Spiral Model

There are three main milestones in our development model; each assigned a full version number. Development of one version is not dependant upon its previous version, but its integration into the final project is. EG Version 3.0 cannot function within the program unless versions 2.0 & 1.0 are complete. To this effect, these milestones are when their actual foci are integrated into the main project, and not when we may begin development. Using this scheme we can keep all members busy on separate parts of the project, give continued feedback on all stages of development, and develop the architecture together at once for all facets of the program instead of Version 1.0 dictating the design of all future versions.

### *Version 0.5*

Version 0.5 is our development level. We will design & create the protocols for the timer service & the connections between the tablets. This will not be implemented in the actual program, but outside as separate programs until we are happy with the results (Speed is of major concern, as is accuracy).

### *Version 1.0*

The main focus of version 1.0 is having all of the separate tablets able to talk to each other through the program and select their roles, whether the user takes the role of game creator, joiner & data input, or view. Specific means to this end include:

- Timer service complete.
- Broadcasting of created games implemented.
- Ability to join broadcasted games available.
- GUI layouts for input/view completed.

### *Version 2.0*

Fully functioning data input is the goal of version 2.0. At this stage all data we wish to collect should be implemented through the GUI and stored in the database. Errors & exceptions relating to data input will need to be caught and handled. Specific means to this end include:

- Complete GUI scheme involving listeners & event handlers.
- Implement database architecture.
- Growth of communication abilities between the tablets. (more messages)

### *Version 3.0*

This version will complete the required part of the project. It is focused on implementing all GUI's involving view of the data. Specific means to this end include:

- Create SQL statements that get the appropriate information.
- Test different timing schemes to auto-query the database.
- Override hot-keys on tablet pc to alternate views as desired.

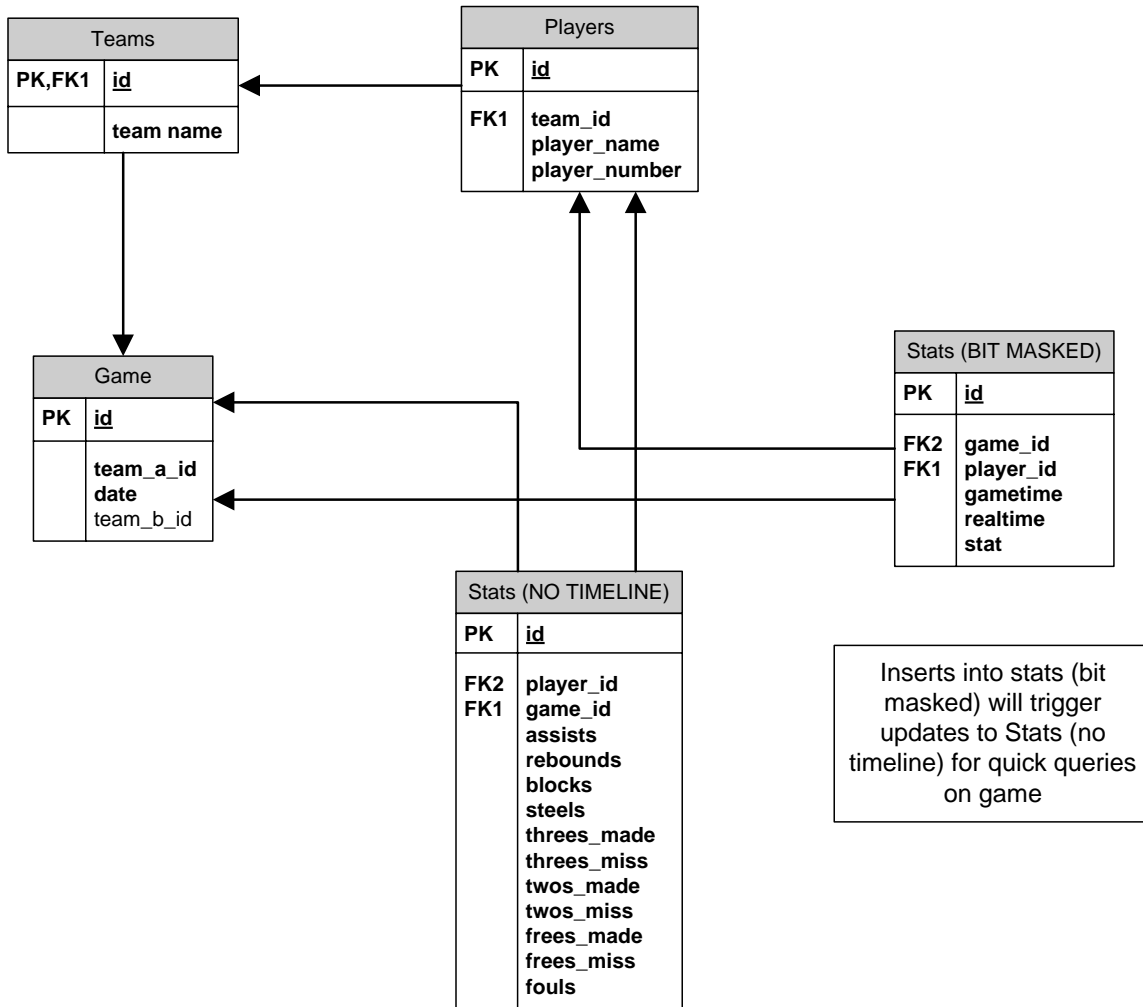
*Beyond version 3.0*

Although version 3.0 completes the requirements for the project, as time permits additional features will be implemented. These include:

- Revisions to all GUI for efficiency and accuracy.
- Database Replication.
- Disaster Recovery (Game Creator computer crashes).
- More advanced error handling (Unique situations that may arise EG Only 3 Player Available).
- More advanced data input: Shot charts, Play calling & grading of ran plays.

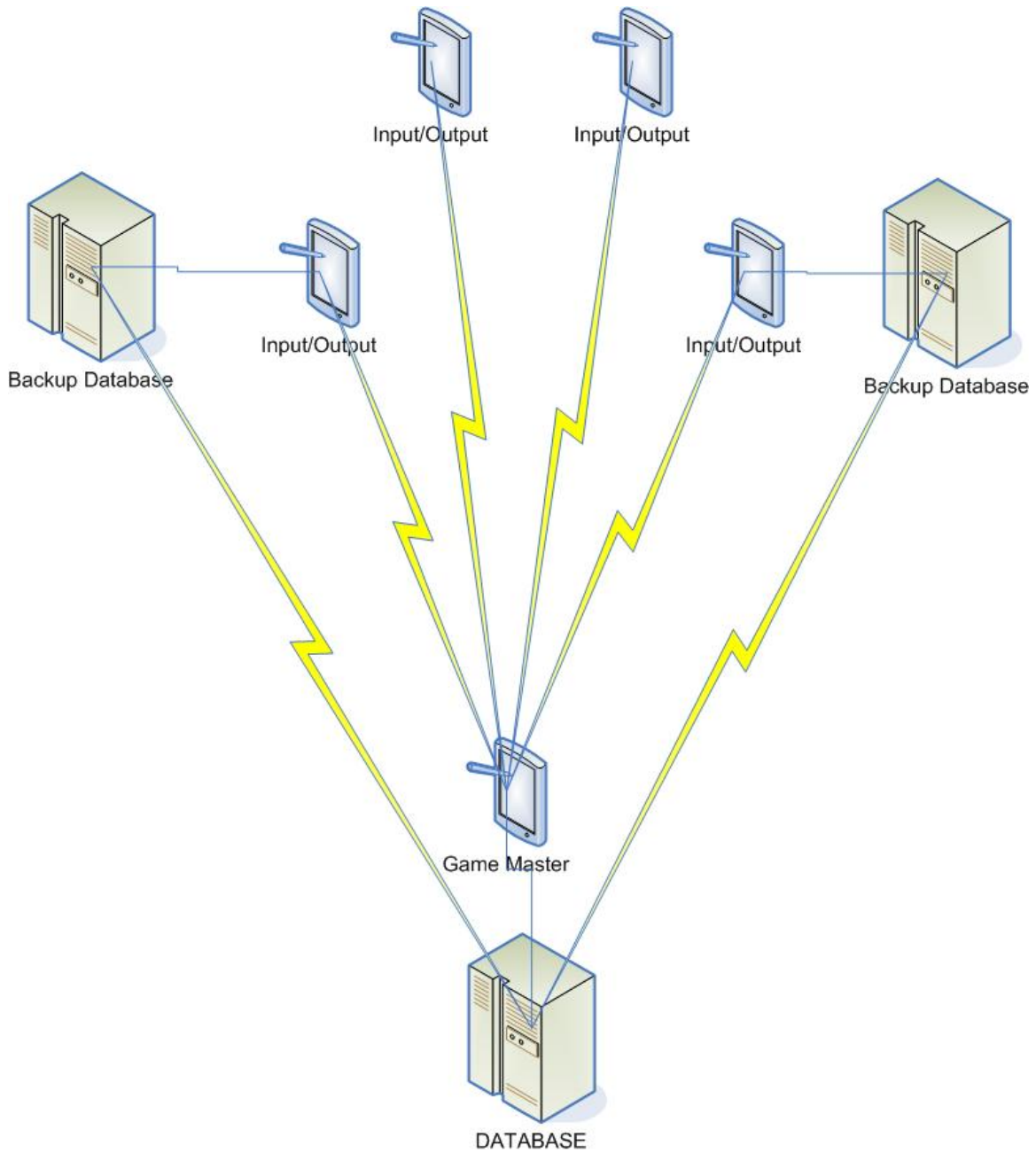
# Known Data Models

## Database Model

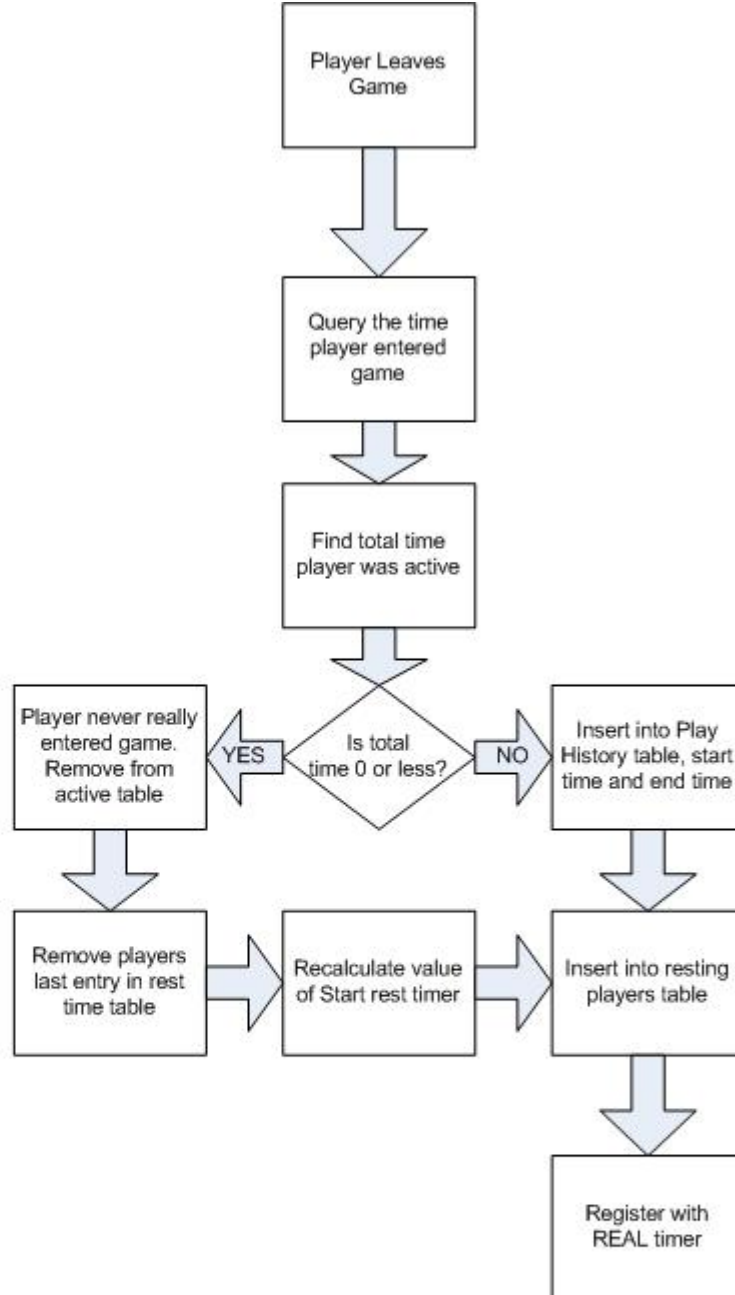
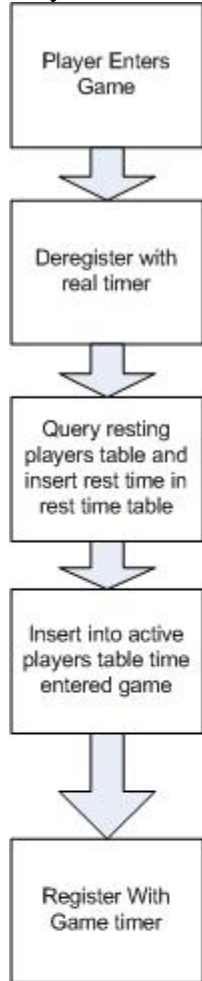


## Database Replication

The game creator has the main database, and every other game joiner will run a backup replication database.



## Database Control Player in-out Control



Queries for current rest time can be calculated by "SELECT (NOW() - START\_TIME) as total FROM `resting\_players`" and adding to total as real time continues.

Likewise for active time

## Timer System

