|  |
| --- |
| Neelima Saraf and Co |
| Tournament tracker |
| NCA Tournament tracker |

|  |
| --- |
| Windows User  11/10/2021 |

Table of Contents

[Product vision 3](#_Toc87446218)

[Project overview 4](#_Toc87446219)

[Design decisions 5](#_Toc87446220)

[Operations documentation 6](#_Toc87446221)

[Requirements documents 7](#_Toc87446222)

[Support documentation 8](#_Toc87446223)

[System documentation 9](#_Toc87446224)

[User documentation 10](#_Toc87446225)

# Product vision

Essence of a product : Tournament tracker is an application that setups the schedule for team to play each other in a single-elimination style matchup.

Useful for:

* Office ping pong tournament
* Rec league playoffs
* 3v3 basketball league etc.

This document is specific for NCAA basket ball tournament.

Summary of the current cost estimates

Predicted benefits

Risks

Staffing estimates

Scheduled

# Project overview

This is a summary of critical information relevant to the project,

primary user contacts,

technologies - Windows form based application

tools used : Visual studio 2019, sql server 2019

# Design decisions

This is an overview of critical decisions related to design and architecture that the team made throughout the project. Document your design decisions throughout other artifacts, such as system overviews and source code.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Questions** | **Answers** | **Additional paints** |
|  | How many teams will tournament handle, Is total team variable | Yes |  |
|  | Should tournament store information about the team members |  |  |
|  | If a tournament has less than the full competent of players, how to handle it | At tournaments with less than perfect number (a multiple of 2) should add in ‘byes’. Basically, certain people selected at random get to skip the first round and act as if they won |  |
|  | Should opponents be chosen random or any specific order | Random |  |
|  | Should we schedule a game or they play whenever | The game should be played in whatever order and whenever the players want to play them |  |
|  | If the games are scheduled. How does the system know when to schedule game for | Not scheduled. So does nto matter |  |
|  | If games are played whenever, can games from second round be played before first round is complete | Second round will be played only when first round is completed |  |
|  | How to determine who won the game |  |  |
|  | Do we need to store team points | Winner will have 1 and looser will be 0 |  |
|  | Do we need to store numbers of players in each game |  |  |
|  | Who are the users of the application and what access level do they have | No level of access |  |
|  | What kind of front end does this system has | Desktop as of now but web or mobile later |  |
|  | Where will data be stored | MS SQL db but option to store in text file |  |
|  | Will the system handle fees, prize and payouts | Yes, The tournament should have the option of charging an entry fee.  Prizes should also be an option, where the tournament administrator chooses how much money to award a variable number of places. Total cash amount should not exceed income from the tournament. A percentage based system would also be nice to specify |  |
|  | What type of reporting is needed | A report specifying outcome of the games per rounds as well as a report that specifies who won and how much they won. They can be dilapyed on a form or emailed to tournaments competitors or administrator |  |
|  | Who can fill in the results of the game | Anyone using the application can use the system |  |
|  | Should this system communicate users/team about upcoming events | Yes, email to user(team members) about round as well as schedule |  |
|  | Is each player are there on their own or system can use tournament tracker | Yes more members can be added. Teams should be able to name their teams |  |
|  |  |  |  |

* **Structure: Windows Forms applications and class library**
* **Database: SQL and/or Text file**
* **Users: One at a time on one application**
* **Key concept** 
  + **Email**
  + **SQL**
  + **Custom Events**
  + **Error handling**
  + **Logs**
  + **Interfaces**
  + **Random order**
  + **texting**

**Database design**

**Team**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Teamid** | **TeamName(string)** | **TeamMamber(List<person>)** |  |  |  |
|  |  |  |  |  |  |

**Person**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FirstName (string)** | **LastName(string)** | **EmailAddress(string)** | **CellPhoneNumber(string)** |  |

**Tournament**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TournamentID**  **(string)** | **TournamentName**  **(string)** | **Fee**  **(decimal)** | **EnteredTeams**  **(List<Team>)** | **Prizes**  **(List<Prize>)** | **Rounds**  **(List<List<Matchup>>** |  |

**Prize**

|  |  |  |  |
| --- | --- | --- | --- |
| **PlaceNumber(int)** | **PlaceName(string)** | **PrizeAmount(double)** | **PrizePercentage(double)** |

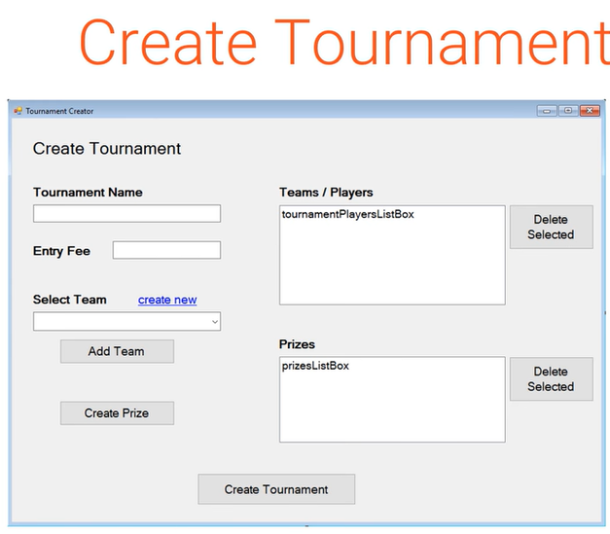
**Matchup**

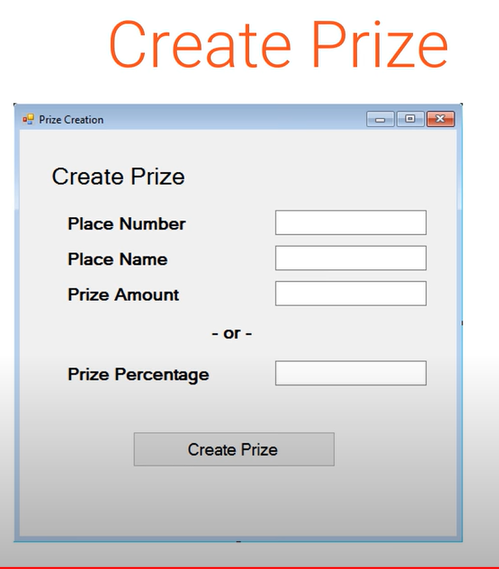
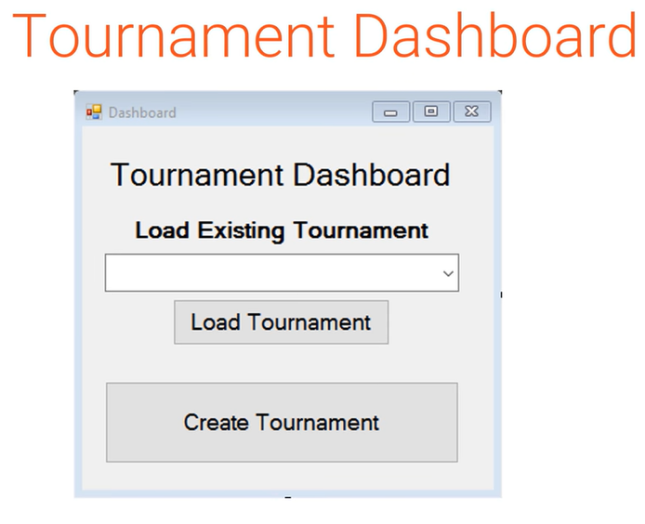
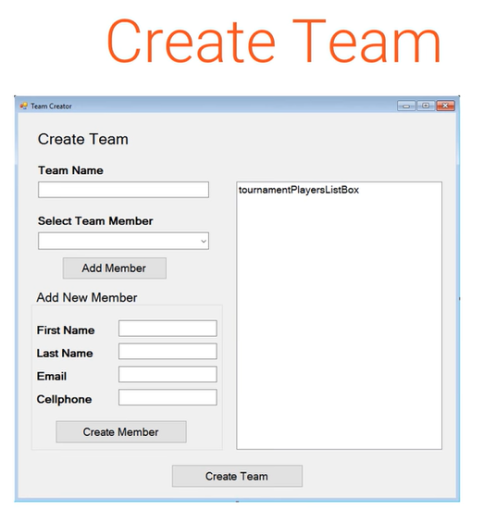
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Entries(List<MatchupEntry>)** | **Winner(Team)** | **MatchupRound(int)** |  |  |  |  |

**MatchupEntry**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TeamCompeting(Team)** | **Score(double)** | **ParentMatchup(Matchup)** |  |  |

**Pages**

******

******

# Operations documentation

This is usually a description of the dependencies that your system is involved with, references to backup procedures, troubleshooting guidelines, etc. Your operations department likely has a standard format for this type of documentation.

# [Requirements documents](https://www.nuclino.com/articles/product-requirements-document)

It's an overview of what the system does, including use cases, user stories, or essential user interface prototypes, etc. Aim to capture the requirements as executable specs.

# Support documentation

This includes the training materials specific to support staff, trouble-shooting guides, etc. Like the operations department, the support team may have standard templates or examples that you can work from.

# System documentation

Provide an overview of the system, including technical architecture, business architecture, and high-level requirements. It helps ensure that if the development team leaves, critical information is left behind.

# User documentation

t includes reference manuals and support guides for the users. Keep it simple and easy to understand. the solution design is flawed if you need to extensively train your users on how to use it.