



IFM03A3 || IFM3A10

INFORMATICS 3B

MINI PROJECT

LEAGUE MANAGEMENT

OVERVIEW

The Informatics 3B individual mini-project serves as a replacement for the main Third Year Project. This option is only available to students that did not pass the practical component of Informatics 3B i.e. IPM (Individual Project Mark) was less than 50%.

REQUIREMENTS

The goal of the mini-project is to supply you with an opportunity to demonstrate to us your basic understanding of the fundamentals of control structures and problem-solving techniques.

The following criteria are applicable regardless of the project you choose:

- The project must be a Client-Server application targeting a small user base (i.e. a small business)
- The project must have a web-enabled component
- The project must have a mobile application
- A desktop (windows forms) component (if applicable)
- The application must include data access with the following conditions:
 - Create, Retrieve, Update and Delete functionality as appropriate
 - Your application may use any RDBMS except MS Access

DATES

You will have three demonstrations for your mini-project for which you must make an appointment on the following dates:

DELIVERABLE	DUE DATE	PRACTICAL COMPONENT WEIGHT
Alpha Demonstration	11 September 2020	10%
Beta Demonstration	9 October 2020	20%
Final Demonstration	30 October 2020	70%

DELIVERABLES

ALPHA DELIVERABLE

The alpha version of the system is a partially completed prototype of the final solution. At a minimum, students are expected to have the following elements operational at the demonstration:

- Graphical User Interface – The user interfaces to the system should be built, but not all functionality need to be associated yet. However, functional navigation must exist between all user interfaces i.e. you must be able to navigate between interfaces.
- Basic System Functionality – Basic data manipulation and handling functionality should be present in all applications.
- Database connectivity and operations – The system should be able to handle communication and basic operations (Add, Edit, Update, Delete, etc.) with the database system used.
- Database relationships – The database designed in previous deliverables should be implemented in a relational databases system.
- Demonstrate and / or prove that the system being developed adds value to the solution and solves a problem.
- Source Control and Deployment

BETA DELIVERABLE

The beta version of the system is a nearly completed prototype of the final solution. At a minimum, students are expected to have the following elements operational in addition to the functionalities demonstrated in Alpha:

- Working functional solution – All functionalities represented by the use cases identified in previously has been implemented.
- Complete and working user interfaces, on all applications, which are easy to use i.e. user friendly and well designed in terms of the layout and colour choices.
- Functional Reporting on both Operational and Management Levels.
- All agreed upon changes as discussed with the project team by the project mentor and external examiner in the alpha have been implemented.
- Source Control and Deployment

FINAL DELIVERABLE

A 5-minute project overview in PowerPoint.

- The complete project and project features in 15-20 minutes to marker(s).
- A logical, structured demonstration of the final working system.
- Justification that the interface to the system, the utility of the system, the algorithmic and technical complexity of the project is satisfactory.
- System must incorporate realistic data in the database.
- Entire system must be deployable
- Use of source control must be demonstrated
- Students are subjected to a questions and answer session by the assessor(s)



PROJECT DESCRIPTION

You are required to implement a **league management system** for any sport of your choice. The **league owner** must be able to **create a new league** of the specific sport. **Teams** are created and added to the league after which **players** are created and added to teams. **Venues** can be added to the league where games will take place.

The league management should then be able to create or **generate fixtures**. A fixture will involve the **two teams who will play against one another and the venue** at which the game will take place. A game takes place on a certain date and at a certain time.

During the creation of fixtures, adding of players to teams, etc. the system should ensure that no double allocations are taking place. For example, two fixtures cannot take place at the same time at the same venue.

When a specific match has been played, the manager must add the scores from this specific match. Scores for each fixture must be added to the specific player that got the score. Once scores have been added, a **post-match report** should be generated. The league management system must then automatically update the **log table** for the specific league.

The log table must be viewable at any time during the league.

USE CASES

