

Platinum Analytics

Final Year Project Report

DT228

BSc in Computer Science

**Peter Collins**

**<Supervisor>**

School of Computing

Dublin Institute of Technology

**<Date>**



Abstract

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<Student Name>

<Date>

Acknowledgements

Body text

Table of Contents

*(These are suggested contents only. You can add or modify if required)*

1. Introduction
   1. Overview of the project and the background behind it.
   2. Project Objectives
   3. Project Challenges
   4. Structure of the document
2. Research

Current solutions

Algorithm Research

* 1. Eg. Research related to identifying the problem that this project solves, research into solution definition
  2. An overview of the technologies evaluated and selected or rejected and the rationale behind the key decisions.

1. Design
   1. Identification of a design methodology including why it was chosen
   2. Design of each of the project components eg: the UI, Network, Project Demonstration, source code layout
   3. Clearly identifying the list of features and use cases supported within the project.
2. Architecture & Development
   1. Overview of the system architecture and a diagram to represent all of the key elements within the architecture.
   2. Details of each component within the project, problems encountered and resolved, challenges overcome or worked around.
   3. Identify key development components;
   4. Identification/explanation of external APIs used versus own code ; List of classes of your code etc .
3. System Validation
   1. Testing
      1. What testing was performed, why it was selected and what are the key use cases within the project.
   2. Demonstration
      1. Identify what features can be demonstrated and show screen shots or reference a video online to show the project demonstration (for audience not at demo)
4. Project Plan
   1. Project Plan analysis and review of how it changed from the initial proposal including explanation of what changed and why, and suggestions on how to address this if the project was repeated.
5. Conclusion
   1. Analysis of the projects key elements identify the key learning obtained from the project and recommendations and suggestions for how the work can be improved on continued into the future.
6. Bibliography
7. Appendix

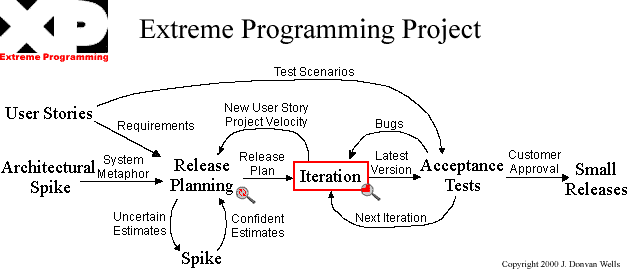
A research sprint to gauge the risk attached to developing the receivers for Android earlier on in the project would have shown that the added

A Research Sprint earlier on in the project would have shown that using an Android Phone as a receiver would result in

## 3. Project Design and Methodology

METHODOLOGY

For this project I used an agile methodology. The Agile Process used was Extreme Programming. According to this definition from extremeprogramming.org, Extreme Programming "is a lot like a jigsaw puzzle. Individually the pieces make no sense, but when combined together a complete picture can be seen."



PROJECT REQUIREMENTS

High level project requirements.

USER STORIES

PROJECT RISK

PROJECT COMPONENTS

This chapter will cover the initial design elements of the system and it's architecture. These designs may be subject to change as the projects grows and evolves.

This system uses a Three-Tier Architecture comprising of a Presentation Tier, Middle Tier and Data Tier. Three-tier is a client-server architecture pattern in which the user interface, business/functional logic and data storage are developed as individual modules on separate platforms. (3) This modular application makes it possible to replace, update or upgrade any tier independently of the other tiers.

**Presentation Tier - The Client**

This is the User Interface for the system and in a web application it is usually the client's web browser. This is the tier from which the user interacts with the system. The user may view content, log in, request files or data and fill in forms. All of these request are sent to and handled by the Middle Tier.

**Middle Tier - Application Server**

This tier handles all of the logic and controls the functionality of the system. Any request from the user are processed here with validation and authentication being performed. Requests for data are taken from the user are received here, sent to the data tier and the response generated and returned to the user. This tier is sometimes referred to as middleware.

**Data Tier - Database Server**

This tier is concerned with the maintaining and serving of all data required by the system. The data may be stored in a shared folder but is more commonly stored in a database, especially for web applications. This tier usually provides a means of exposing the data to the Application server as well as a way to manage it which may be done by making an API available.

### PROJECT COMPONENTS

This project uses an N-Tier application architecture. Each layer is specified below.

SENSORTAG - BLUETOOTH SENSOR

The sensor worn by the player which will be used to measure players' positions on the pitch at all times.

PNM-MON - BLUETOOTH RECEIVERS

This is the python application which runs on the raspberry pis. Its purpose is to capture and store analytics taken from the players.

Python Modules

-bluetooth

-commandline "sudo hcitool lescan" to scan, "sudo btmon" to read input

-datamanager

PNM-DB - REST API / DATABASE SERVER Persistence Level

Once the data is collected and on the Raspberry Pis it is synced to the database for permanent storage via the Rest API. pnm-db is built in python using the Django Rest Framework, with a Postgres Relational Database.

PNM-WEB - FRONTEND APPLICATION SERVER

Built using Django, pnm-web is the frontend web server from which users can view all player analytics

DJANGO

[Django](https://www.djangoproject.com/) is a framework consisting of several tightly coupled elements https://www.djangoproject.com/. Django is an MVC-style framework

PYTHON

Why python?

" First, the use of indentation reduces visual clutter and makes programs shorter, thus reducing the attention span needed to take in a basic unit of code. Second, it allows the programmer less freedom in formatting, thereby enabling a more uniform style, which makes it easier to read someone else's code." Guido van Rossum  
Reston, VA, May 1996