



## Assessed Coursework

Course Name	Web Application Development 2		
Coursework Number	4 (of 5) – Group Project Web App		
Deadline	Time: 4.30pm	Date: 28 March 2025	
% Contribution to final course mark	25	This should take at most this many hours:	20
Solo or Group <input checked="" type="checkbox"/>	Solo	Group <input checked="" type="checkbox"/>	
Submission Instructions	Via Moodle – see Page 4		
Who Will Mark This? <input checked="" type="checkbox"/>	Lecturer	Tutor <input checked="" type="checkbox"/>	Other
Feedback Type? <input checked="" type="checkbox"/>	Written <input checked="" type="checkbox"/>	Oral	Both
Individual or Generic? <input checked="" type="checkbox"/>	Generic	Individual <input checked="" type="checkbox"/>	Both
Other Feedback Notes			
Please Note: This Coursework cannot be Re-Done			

### Code of Assessment Rules for Coursework Submission

Deadlines for the submission of coursework which is to be formally assessed will be published in course documentation, and work which is submitted later than the deadline will be subject to penalty as set out below. The primary grade and secondary band awarded for coursework which is submitted after the published deadline will be calculated as follows:

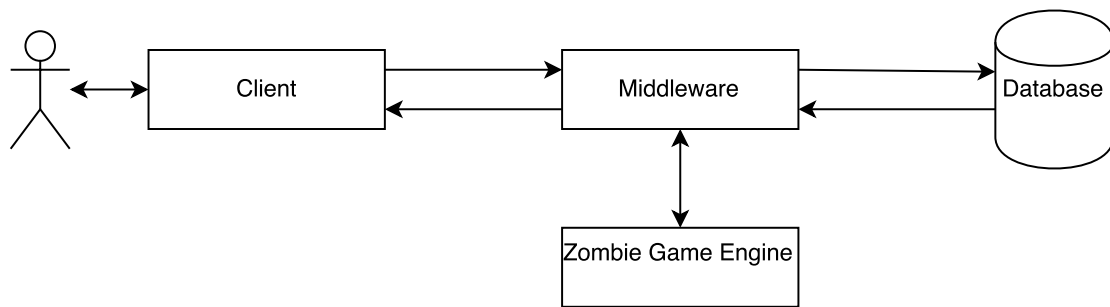
- (i) in respect of work submitted not more than five working days after the deadline
  - a. the work will be assessed in the usual way;
  - b. the primary grade and secondary band so determined will then be reduced by two secondary bands for each working day (or part of a working day) the work was submitted late.
- (ii) work submitted more than five working days after the deadline will be awarded Grade H.

Penalties for late submission of coursework will not be imposed if good cause is established for the late submission. You should submit documents supporting good cause via MyCampus.

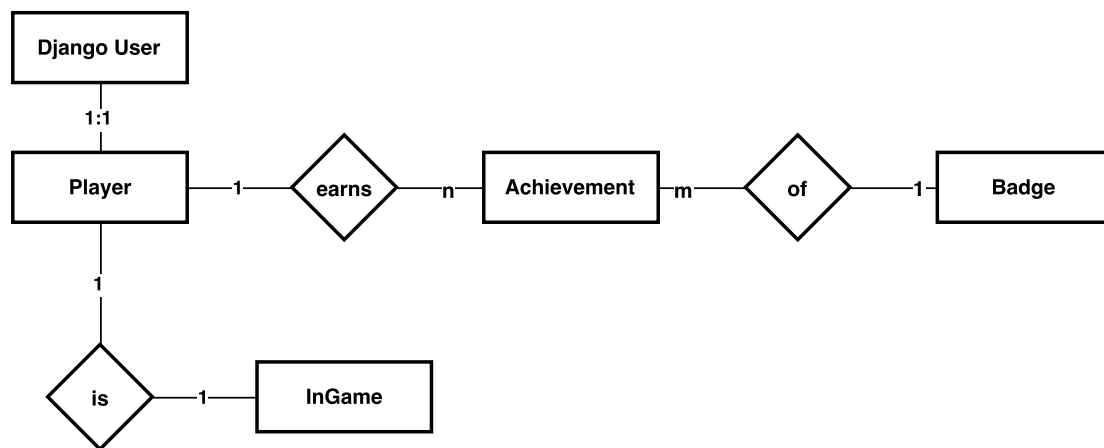
**Penalty for non-adherence to Submission Instructions is 2 bands**

### Marking Criteria

See Page 5



Here is an ER diagram:



We assume that a Player can only play one game at one time. Statistics on pasts games are recorded in the Player model.

Since they can only be in one game at one time, we can store the game in the player model, too.

**Player**(user, profile\_picture, games\_played, most\_days\_survived, most\_kills, most\_people, current\_game)

username, password, email, are stored in the django user model.

Current\_game stored the current instantiated game (tip: use the python pickle package to persist the game object, and save it in the Player model).

How the **Badge** is represented depends a lot on the implementation. My suggestion is below, but other justifiable versions are acceptable, especially if they work in an elegant fashion.

**Badge**( name, description, criteria, badge\_type, level, icon)

Where *badge\_type* (kills, people, days) and *criteria* is an integer specifying how many kills, people, days, etc have to be obtained in order to get the badge.

*Icon* is an image that represents the badge. The population script should ensure that the badges and their icons are all inserted into the database.

A class/handler is needed to determine if the *criteria* for a particular *badge\_type* is met, or not.

Achievement (player, badge, date\_awarded)