






| Mark range % | 100-80% | 79-60% | 59-40% | 39-20% | 19-0% |
|--|--|---|---|--|---|
| Assessment Criteria | "A system ready to deliver to go live"  | "A usable system with useful features"  | "A minimum viable system"  | "Basic data/items listings"  | "Non-functioning system"  |
| Well designed OO code using MVC Design Pattern, installed on university servers (20 Marks) | Considerable amount of OO code (including inheritance) demonstrating correct use of MVC, classes/properties and methods used with appropriate names, using PDO to access the database. Excellently commented code. Submitted code presented organised, neatly and readable and excellently commented. Demonstrable performance enhancements. | Reasonable amount of OO code demonstrating correct use of MVC, classes/properties and methods, using PDO to access the database. Well commented code. Implementation of performance enhancements. Submitted code presented organised, neatly and readable and well commented. | Small amount of OO code demonstrating correct general use of MVC, classes/properties and methods, using PDO to access the database. Some code comments. Consideration of performance. Code submitted and organised. | Small amount of code, not OO, demonstrating limited understanding of MVC, classes/properties and methods, or database access. Basic code comments. Code submitted. | Very little code demonstrating no real understanding of MVC, classes/properties and methods, or database access. Little or no code comments. Significant amounts of copied code, poorly presented. |
| Database | Installed and running on the university web server and using university database server. | | Install on the university server but might have some issues running. | Not installed on the university web server and and/or not using university database server. | |
| Ability to create self-registration, records and login (20 Marks) | Users can register themselves and login successfully using encryption for their passwords. Anti-spam feature used. Some form of session used to maintain state. Malicious code filtering on items. Driven by excellent, logical OO design. Excellently commented code. Test accounts working. | Users can register themselves and login successfully using encryption for their passwords. Some form of session used to maintain state. Validation on input fields. Driven by good, logical OO design. Well commented code. Test accounts working. | Users can register themselves and login successfully. Some form of session used to maintain state. Basic OO implementation. Some code comments. Test accounts working. | Registration and login not fully implemented/working correctly. Test accounts not fully working. Basic code comments. | Registration and login not implemented, or not working. No test accounts or not working correctly. Significant amounts of copied code, poorly presented. |
| Ability to display information from records stored in the database (20 Marks) | Rental users can retrieve and display user details appropriate to their role, including images. Sophisticated, responsive layout using CSS/Bootstrap. Includes paging and web scraping protection. Demonstration of system with large number of realistic records/users. Driven by excellent, logical OO design. Excellently commented code. | Rental users can retrieve and display user details appropriate to their role, including images. Good responsive layout using CSS/Bootstrap. Demonstration of system with large number of records/users. Driven by good, logical OO design. Well commented code. | Rental users can retrieve some item details. Basic cellular listing layout using tables or divs with some formatting evident. Attempting display of individual items. Basic OO implementation. Some code comments. | Rental users can retrieve some item details. Simple table or list of text data, Basic code comments. | Users cannot retrieve or display details from the database. Little or no code comments. Significant amounts of copied code, poorly presented. |
| Ability to search for charge points (20 Marks) | Comprehensive, interactive faceted search to narrow down output to less than 10 results using all charge point parameters. Complex SQL statements. Driven by excellent, logical OO design. Excellently commented code. | Free text search facility with 2 or 3 extra filters combined into the search. Good OO implementation. Well commented code. | Free text search facility working but basic. Some filters implemented, but may have issue. Basic OO implementation. Some code comments. | Basic text search facility may have issues. Basic code comments. | Some search/filters partly implemented or no implementation. Little or no code comments. Significant amounts of copied code, poorly presented. |
| Ability to create/edit charge point (home owner). (20 Marks) | Home owner users can add, view and edit their "charge point". User profile images included. Driven by excellent, logical OO design. Excellently commented code. | Home owner users can add, view their "charge point". User profile image included. Good OO implementation. Well commented code. | Attempt at "charge point" creation and viewing for the home owner user. Not sophisticated/some issues present. A Basic OO implementation. Some code comments. | Basic "charge point" feature, possibly from manually entered data in the DB but has issues.. Basic code comments. | Little or no implementation. Little or no code comments. Significant amounts of copied code, poorly presented. |