|  |  |
| --- | --- |
| **File Name** | **Description** |
| R2 Test.db  R2 Database Entry.docx | This database file was made for two distinct purposes. As the test cases progressed, the requirements became greater and errors in testing would be potentially harmful.  This database was developed to protect the initial database from any errors that occurred in the testing phase, such as the accidental deletion of the entire database’s content. This also allowed the database to test fields such as payment fields without requiring any special permissions.  As this database was disconnected to the inputs of the actual solution, it also served as a backup database in case there were any malfunctions that lost the main solutions database. |
| R2 Notifications.docx | This code excerpt incorporates the push notification system that users of the solution may experience. These notifications are set to occur when a successful payment is made, when there are adjustments to the parking rules, or when an emergency notification is required. Part of the requirement of this system was that the notifications be dismissible as dictated by the user. These notifications were a requirement set by the client team in the second release in order to ensure that users are aware of their situation and can take action if required.  These notifications incorporate aspects of Java, Python and HTML, although for the purposes of artefact evaluation, were contained in a single word document as stated to the left. |
| R2 Test Cases.docx | The python test cases for Release 2 can be seen in the **forms.py** file in the repository. For the purpose artefact evaluation, they were placed into a separate word file as seen on the left.  These test cases use the testing functions designed around database access and control to test the newly developed databases. This includes inserting, updating and removing entries in the Health and Safety, Parking, Other and payment databases. This was done to ensure that all forms of input were compatible with the databases. |
| R2 Test Functions.docx | The python test functions for Release 2 can be seen in the **testfunctions.py** file in the repository. For the purpose artefact evaluation, they were placed into a separate word file as seen on the left.  These functions served a similar purpose as the ones in the last release, i.e. to test the functionality of the solution. As the functions would have required a broad range of test cases, the functions themselves were made to accept a large number of variables. This allowed for fewer functions to incorporate all the test cases. |
| R2 Test Pseudocode.docx | Similar to the pseudocode of the first release, this document has the intent to assist the development of the test cases in the second release.  This iteration of pseudocode was developed in the planning stages of the second release and focused on database entry, payment requirements and notification requirements. As with the last section of pseudocode, this excerpt is structured to display the input from certain forms as well as their expected output.  In order to ensure the relevance of this code for the second release, the hypothetical persona was altered to include payment details and account numbers. |
| R2 In Page Form Validation.docx | Unlike the previous form validation complete in Release 1, these form validations were complete using **python** code using the functions created in **testfunction.py**.  This validation ensures the input for the **Health and Safety** report page and the **Visitor enquiry** information page. This includes ensuring correct email format, numerical or alphabetical input, as well as general valid inputs.  These checks ensure that the only inputs accepted into the database are valid variables which correspond to the required functionality. E.g. ensuring no conflicting information is entered. If there is not a valid input, the variable is set to null and not allowed to be passed. |