

Test Specifications

for

Sydney Wildlife Rescue and Care App

Version 2.0 approved

Prepared by Samuel Hickman

Macquarie University PACE Group 9

27 May 2016

Table of Contents

Revision History.....	2
1. <u>Testing Strategy</u>	<u>3</u>
2. <u>Functional Test cases</u>	<u>3</u>
3. <u>Non-functional Test cases</u>	<u>11</u>
4. <u>Test plans</u>	<u>12</u>

Revision History

Name	Date	Reason For Changes	Version

1. Testing strategy

In order to test our product we intend to run both black box and white box testing of our product, one which will be used to purely track the input and output of the project without looking at how the data is being manipulated, and white box testing which will be looking out how data travels and is transformed through the system. We will also incorporate end user testing if possible in order to identify faults in the system from an end user perspective. Our testing schedule will mainly take place after week 9, as this is when we expect to have a testable system in place. White box testing will be done as early as week 7, as it can be performed on very earlier prototypes.

White box testing will be handled in weeks 7-13.

Black Box testing will be handled in weeks 9-13

End User testing will be handled in week 11-13

Some of the testing resources that we have assigned include the various software design platforms that we are using to develop the software solutions, such as unit testing in eclipse, etc. as well as end users for end user testing. We will also use manual data inputs as testing resources, and custom data designed to find faults and test the system in its various capacities.

Our testing milestones will be based around the complete testing of each subsystem, where when each subsystem is completely tested based on our test case specifications, this will signify a milestone representing that this subsystems testing is complete. Once all subsystems have been completely tested, and testing for the system working in conjunction is complete, this will be the final milestone of the testing process.

Our test deliverables will be the success of the various tests assigned to each subsystem of our overall system, as well as various testing documents or artifacts such as test metadata, testing results/reports, summary reports etc.

2. Functional test cases

Test Case ID: 1.1

Test Case Description: The test will evaluate if and how a new animal can be entered in to the system, including profile creation, and whether this subsystem is functional

Requirement(s) Tested: R1, R1.1, R1.1.1, R1.2, R1.6

Pass Requirement: The creation of an animal is successful, and is properly recorded in to the system. The animal's profile can then be viewed and manipulated properly

Fail Requirement: The creation of the animal is unsuccessful in anyway, and/or is not properly recorded in to the system.

Test Data Input: The test data input for this test will be the information necessary to create a new animal profile, which is given by the e-form for animal profile creation. As an example, it includes fields such as species, record number, carer who found it, etc. Valid input data will include a proper animal profile input, while invalid data input will be an invalid animal profile.

Test Data Output: The data output of this test should be a record within the database that records all required information for the animal, as well as the system for it to create the animals profile.

Test Case ID: 1.2

Test Case Description: The test will evaluate whether the e-form for entering a new animals information in to the system for profile creation has all necessary fields as specified by the system requirements, as well as the correct data types and restrictions for each field.

Requirement(s) Tested: R1.3, R1.4, R1.4.1, R1.11, R1.12

Pass Requirement: The e-forms fields are all consistent with the requirements as set forth by this document as well as the sponsor, and data entry is restrictive and consistent with data field types (E.g, record number should always be an integer, never a float or some form of string).

Fail Requirement: The e-forms fields are not consistent with the requirements, and data entry fields do not restrain input data to their data types.

Test Data Input: The test data input for this test will be the comparison of the e-form to the requirements, as well as the input of data conforming to the proper fields data types, as well as data that is incorrect for that field's data type. Valid data input will be a complete e-form or correct data, while invalid input data will be an incomplete e-form or incorrect data.

Test Data Output: The output for this test will be the results of the analysis. This will either be a pass where the fields all conform to the requirements, or a fail if they do not, and so is a binary output. If data is entered that conforms to field data types, the e-form should accept the data and raise a success flag to the user. If they do not, the e-form should flag these fields and notify the user of where the errors are occurring, and the system should not accept the input from the e-form.

Test Case ID: 1.3

Test Case Description: The test will evaluate whether the form for entering new animals in to the system is secure according to the security requirements of the system (e.g. does not open any vulnerabilities to the system), and only accessible while logged in as a user of the system.

Requirement(s) Tested: R1.5, R1.10, R1.13, R10

Pass Requirement: The system does not allow access to the e-form while the user is logged off, or if an entity that is not a user tries to directly access the data entry page.

Fail Requirement: The system allows access to the e-form while a user is logged off, or is accessible by an entity that is not a user.

Test Data Input: The input data for this test will be the attempted access of the e-form while logged in, and while logged off. It is a complex data type, but will be a user access form of input data. Valid input data will be a user using the e-form with sufficient privileges, while invalid input will be a user with insufficient permissions or an external entity using the e-form.

Test Data Output: The output data will be a binary pass/fail. If the system allows access while logged in, and does not allow access while not logged in, the test will return a pass. Vice versa, if the system does not allow access while logged in, or allows access while logged off, the test will return a fail.

Test Case ID: 1.4

Test Case Description: The test will evaluate whether the e-form for data entry can be left partially incomplete and can be saved, and then resumed and completed at a later date.

Requirement(s) Tested: R1.7, R1.8

Pass Requirement: The system allows for e-forms to be left partially incomplete and saved, and then resumed and completed at a later date.

Fail Requirement: The system does not allow for e-forms to be saved, and/or partially complete e-forms are unable to be resumed and completed at a later date.

Test Data Input: The test data input for this case will be that of an e-form being entered in to the system. Valid data will be an incomplete e-form entry.

Test Data Output: The test data output will be a binary pass/fail. If the system allows for the e-form to be left partially incomplete and saved, and then resumed later, the test will return a pass. If any of the above pass conditions are not met, the test will return a fail.

Test Case ID: 1.5

Test Case Description: The test will evaluate whether carers can view and search the animals profiles for which they are currently caring for.

Requirement(s) Tested: R1.9, R3.3, R11

Pass Requirement: The carer is able to browse through the animal profiles for the animals they are currently responsible for, and the profiles they can view/search are exactly theirs and no others.

Fail Requirement: The carer is unable to browse through the animal profiles for the animals they are currently responsible, and/or the animal profiles they are able to see do not belong to them at the present time.

Test Data Input: The test data input for this will be the assigning of several different animals through the system. The user will then attempt to view and search for the animals they know have been assigned to them. Another group of animals will be assigned to another user. Valid input data will be a correct search for some animal that exists, while invalid input data will be a search for an animal that does not exist.

Test Data Output: The test data output will be a binary pass/fail. If the user is able to search/view all the animal profiles that have been assigned to them, and only the animals that have been assigned to them and not the other user, the test will return a pass. If any of the pass conditions set above are not met, the test will return a fail.

Test Case ID: 1.6

Test Case Description: The test will evaluate whether the system properly links to the other third party system "Wild Apricot" in order to perform functions such as carer profile searching, as well as viewability of administrators.

Requirement(s) Tested: R2.1, R2.2, R2.3, R11

Pass Requirement: The system successful and proficiently links through to wild apricot, either using a form of hyperlink or using direct public API/application server integration to the SWRCA system.

Fail Requirement: The system unsuccessfully links through to wild apricot, or does not integrate the public API/application server (if available) in to the SWRCA system.

Test Data Input: The test data input for this test will be that of a user search through the SWRCA. Valid data will be a search for an entry that exists, while invalid data will be a search for an entry that does not exist.

Test Data Output: The test data output will be a binary pass/fail. If the user can successfully search other users either through navigation through SWRCA to wild apricot, or directly through SWRCA, and administrators through either of these methods have viewability of all users under the system, then the test will return a pass. If any of the above pass conditions for this test are not met, then the test will return a fail.

Test Case ID: 1.7

Test Case Description: The test will evaluate whether the system correctly links to the third party software “batchgeo” that is used by sydney wildlife for location tracking services.

Requirement(s) Tested: R3.1, R3.2

Pass Requirement: The system should correctly link to the batchgeo system, primarily through hyperlink or public api/application server in order to fulfill location requirements that is not required to be covered by the SWRCA system.

Fail Requirement: The system does not correctly link to the batchgeo system, or does not integrate it in an intuitive manner.

Test Data Input: The test data input will be the end user use of the SWRCA system to navigate to batchgeo. Valid and Invalid data are non applicable in this test.

Test Data Output: The test data output will be a binary pass/fail. If the user is able to successfully navigate to the batchgeo system, or use it within the SWRCA system, as well as use it in an intuitive fashion, then the test will return a pass. If any of the pass conditions for this test data output are not met, the test will return a fail.

Test Case ID: 1.8

Test Case Description: The test will evaluate whether the decision tree meets all requirements under R4 for the decision tree, including decision tree functionality, final state conditions, score based weighting, etc, and overall provides accurate health triage information that correctly predicts and answers queries based on animal health.

Requirement(s) Tested: R4.1, R4.2, R4.3, R4.4, R4.5, R4.6

Pass Requirement: The decision tree is succinct, it fulfills all requirements set under requirement 4.0, and overall is accurate in representing decisions based on real time animal health conditions. It should also reach an outcome that is extremely accurate and correct due to the sensitive nature of health information, and provide an informative recommendation based on simplistic decisions.

Fail Requirement: The decision tree fails any of the requirements set out in requirement 4.0, is not succinct, and/or does not provide accurate questions, and/or does not provide an accurate outcome with an accurate recommendation given that animals current condition.

Test Data Input: The test data for this will be the navigation of the decision tree of a user given three different examples of animal triage. The first example will be that of an animal

in relatively good condition, the second being an animal that is currently in a moderately stable condition, and the final example will be that of an animal in critical condition. The questions and answers along the decision tree will be recorded, and will be checked if they are accurate (e.g. If a pain assessment question is provided, the question should provide accurate examples of pain, such as is the animal grinding teeth, visibly in pain, etc, and answers should cover all possibilities of pain types, such as low-moderate pain/high accessibility to medication, etc.). Valid data will be the navigation of the decision tree using the correct choices given a situation, while invalid data will be the navigation of the decision tree using the incorrect choices given a situation.

Test Data Output: The test data output will be whether all requirements were passed, the questions and answers during the decision tree were accurate, and the user arrived at the correct final condition for the decision tree (e.g. an animal that is in critical condition should be assessed by the decision tree as being in a critical condition). The test data output therefore will be a more complex analysis of the decision tree, with areas being identified as failing being summarised, and areas that identified as passing being summarised.

Test Case ID: 1.9

Test Case Description: The test will evaluate whether the system is scalable to the use of multiple decision trees, and whether extra decision trees are able to be created easily, and implemented easily in to the system.

Requirement(s) Tested: R4.7

Pass Requirement: The system for the creation of decision trees is intuitive, easily done and the system is able to accommodate the creation and viewing of multiple decision trees.

Fail Requirement: Creating new decision trees in the system is unintuitive/difficult, or requires someone with a very high level skill set. The system does not accommodate the creation and viewing of multiple decision trees.

Test Data Input: The test data will include the creation of new decision trees, and running through the process of creation, and then browsing and using the new decision trees. Valid data for input would be a proper decision tree, while invalid data input would be an incomplete/not properly designed decision tree.

Test Data Output: The test data output for this test would be a binary pass/fail. If the system allowed the creation, viewing and use of multiple decision trees, the test will return pass. If it does not meet one or many of the pass conditions, then the test will return fail.

Test Case ID: 1.10

Test Case Description: The test will evaluate if the information stored by the system, including user data, login data, animal data, etc. is viewable and manageable by administrators and IT support, as well as being compilable.

Requirement(s) Tested: R5.1, R5.2, R5.3, R5.5

Pass Requirement: The pass requirement for this test will be that administrators and IT staff are able to view ALL information stored in the SQL database, including the various tables and relationships, and their properties, and this data is editable, including read, write, edit,

delete permissions. The data should also be exportable and compilable to a CSV file format, in order for viewing outside of the database.

Fail Requirement: The fail requirements for this test include the failing of any pass conditions as stated above, as well as unauthorised access to the database by anyone who is not an administrator or IT staff. The pass conditions will be counted as failure conditions if an entity outside of already named entities can perform any of the pass functions on the database.

Test Data Input: The test data input will be the attempted access of the database, as well as attempts to read, write, edit or delete test data entries. We will also test whether the user can view table properties within the database. The exporting of the database will also be tested, in order to evaluate whether the data exports correctly, and is compiled and in the correct file format. A valid input for this will be access using an administrator/IT support account, and invalid input will be the attempted access and manipulation of the database using a carer account.

Test Data Output: The test will consist of a binary pass/fail. If the admin/IT support account can perform the pass requirements, then the test will return pass. If they are unable to perform any required functions, the test will return fail. Furthermore, if the carer account can perform any of the required functions, then the test will return fail, while if they are unable to access it at all, the test will return pass.

Test Case ID: 1.11

Test Case Description: The test will evaluate whether user accounts are manageable, and can be modified/updated where necessary as denoted by the system requirements (e.g. users should be able to reset their passwords).

Requirement(s) Tested: R6.2, R6.3, R6.4, R6.4.1, R6.4.2, R6.5

Pass Requirement: User accounts are manageable, both by users (to a certain extent, such as password resets) and administrators, and are able to be modified/updated where necessary as denoted by system requirements, either through direct database manipulation or a set of developed administrator tools within the system.

Fail Requirement: User accounts are unmanageable, with basic modifying functionality unavailable to users (e.g. password resets), or database manipulation is impossible. Other fail conditions include modifying data usually only accessible to high level users (E.g. carer can modify other carer details).

Test Data Input: The test data input will be the attempted management of users. This will include the testing of password reset, as well as the testing of direct database manipulation. Valid input data will be a user attempting to reset their password, and an administrator attempting to manipulate the database directly or through administrator tools, while invalid data input will be a user attempting to perform direct database manipulation.

Test Data Output: The test will consist of a binary pass/fail. If the pass requirements are all satisfied, then the test will return a pass, else the test will return a fail. If any/all of the fail requirements are met, then the test will return fail.

Test Case ID: 1.12

Test Case Description: The test will evaluate whether the system is able to accommodate multiple users at any one time, as well as retaining “quality” function during times of high load (e.g. Pages should not take any more than 5 seconds to load server side while 100 users are using the system)

Requirement(s) Tested: R7, R7.1, R8, R8.1, R10

Pass Requirement: The system allows use by multiple users at once, both carers and administrators, and functions at a quality level.

Fail Requirement: The system does not allow use by multiple users at once up to a reasonable amount, and/or does not function at a quality level.

Test Data Input: The test data input will be the use of the system by 20 users at once, consisting of a mix of primarily carers with some administrators. Valid data input for this test will be a mix of 15 carers and 5 administrators using the system, while invalid data will be the same.

Test Data Output: The test data output will be a binary pass/fail. If the system accommodates all 20 users, and functions at a high quality level, then the test will return a pass. If the system does not allow any user to use the system during the test, and/or the system does not function at a high quality level, the test will return fail.

Test Case ID: 1.13

Test Case Description: The test will evaluate whether the system is susceptible to encryption attacks, such as a man in the middle attack or brute force attack.

Requirement(s) Tested: R15, R15.1, R15.2, R17, R18

Pass Requirement: The system is resistant/immune to such attacks, such that it would be infeasible to use such techniques to access the database/webserver.

Fail Requirement: The system is susceptible/vulnerable to these types of attacks, and sensitive information can easily be obtained from the database/webserver using these common attacks.

Test Data Input: The test data input would include tools that simulate such attacks, and a security analysis of the security protocols that exist on the web server and in the database. Valid input data would be tools testing the security of the system, while invalid input data would be an actual attack on the web server/database.

Test Data Output: The output will be a detailed analysis of security, including specific vulnerabilities and what types of common attacks the web server/database is vulnerable to. This would take the form of a report that would be analysed by administrators/software engineers.

Test Case ID: 1.14

Test Case Description: The test will specifically evaluate the susceptibility of the database to specifically the most common attack against an SQL database, which is an SQL injection attack. Because of the nature of SQL injection attacks (malicious sql code inserted in to data fields in order to cause the database to dump information to the attacker), the SQL code behind the database and website, specifically data fields, must be written in a way that conforms with best use standards and SQL injection resistance standards. In built security

software that exists on platforms such as PHPmyadmin and MySQL, as well as external software should also exist in order to combat this attack.

Requirement(s) Tested: R18.1

Pass Requirement: The system is resistant/immune to SQL injection attacks.

Fail Requirement: The system is vulnerable to SQL injection attacks.

Test Data Input: The input data will be the attack of the database using an SQL injection testing tool, such as sqlmap. The valid data will be the attack of the database using this tool, invalid data would be an actual attack on the database.

Test Data Output: The test data output will be a comprehensive report detailing where and why areas of the website/database are vulnerable to SQL injection attacks.

Test Case ID: 1.15

Test Case Description: The test will evaluate whether the system will send notifications to administrators/carers based on when an animal under their care has health (e.g. weight, forearm measurement, etc) that falls below the expected health curve for a recovering animal of that species.

Requirement(s) Tested: R3.4

Pass Requirement: The system will send a notification to the current carer/administrator of the animal when the weight of the animal is recorded as being 5 Kilograms lower than the expected weight at the same age of an animal of the same species.

Fail Requirement: The system fails to send a notification when the pass requirement is met, or sends a notification when the health data does not drop beyond the set threshold of the notification.

Test Data Input: On the animals profile page, a user will record the animal as being a certain amount of Kilograms underweight when comparing it to the standard weight of an animal of a certain age. Valid input data will be 7 Kilograms underweight, while invalid input will be 2 Kilograms underweight.

Test Data Output: The test will return a binary output pass/fail. If, when 7 kilograms underweight, is recorded, and the system sends a notification and if, when 2 kilograms underweight is recorded, the system does not send a notification, the test will return a pass. Failure to meet any of the pass conditions for the test will result in the test returning fail.

3. Non-Functional Test Cases

Test Case ID: 2.1

Test Case Description: The test will evaluate whether the decision tree and website is easily usable on different screen sizes, such as a PC monitor, tablet and mobile phone, and functions on each of these screen sizes.

Requirement(s) Tested: R12.2, R12.3, R12.3.1, R14, R14.1

Pass Requirement: The website and decision tree are viewable on a variety of screen sizes, including common screen sizes for desktop monitors, tablets and phones, are all easily usable and are functional.

Fail Requirement: The website and/or decision tree does not appear properly on a variety of screen sizes, and/or is not functional on one/many screen sizes.

Test Data Input: The test data input for this task will be the use of the website and decision tree on a PC, a tablet and a mobile phone. Valid data for this will include use of the website and decision tree on a variety of screens, while invalid data is non-applicable in this test case.

Test Data Output: The test data output will be a binary pass/fail. If the system is viewable on all different screen sizes, and is functional, then the test will return a pass. If the system fails any of the pass conditions, the test will return fail.

Test Case ID: 2.2

Test Case Description: The test will evaluate whether the database view is accessible (e.g. easily able to be sorted, filtered, and information is easy to glean if looking at the database tables).

Requirement(s) Tested: R5.4

Pass Requirement: The database has all accessibility standards met as per proper database design functionality set out by SQL database authorities. Data is able to be sorted, filtered and information can be looked at without confusion.

Fail Requirement: The database does not meet one or more accessibility standards set out by SQL database authorities.

Test Data Input: The test data input will be an end user assessment of database manipulation, specifically targeted at a database administrator. Valid input data would include the use of the system by a database administrator, while invalid input data would be use by a carer, or other low level user

Test Data Output: The test data output will be a detailed feedback report based on the user testing it. This could take the form of a questionnaire or written statements regarding the use of the database.

Test Case ID: 2.3

Test Case Description: The test will evaluate whether all parts of the system as a whole, as well as each individual subsystem, conform to the code of ethics set out by Sydney Wildlife as well as government authorities on animal control in relation to information systems.

Requirement(s) Tested: R15.3

Pass Requirement: The system does not breach any section of either ethical standards, and meets all mandates and law requirements set out by each of these codes.

Fail Requirement: The system breaches any one or more code of ethics set out in either of the ethical standards, and/or does not meet all ethical mandates and laws set out by Sydney Wildlife or any associated authority.

Test Data Input: The test data input will be a detailed analysis of all systems and subsystems of the SWRCA system in relation to ethics, and will compare each and all parts of the system to both ethical standards. Valid and Invalid input data are non applicable in this situation.

Test Data Output: The test data output will include a report detailing each part of the system, and how it relates to each ethical standard. Areas where it breaches any ethical/law mandate will be identified, so that they can be immediately corrected.

4. Test plans

4.1.Tools/Resources

Some of the tools we will be using for specific testing will include PHPmyadmin, MySQL, Apache Server, XAMMP server, Dreamweaver, Google Chrome, Eclipse, SQL Inject me, SQLmap. The tools PHPmyadmin, MySQL, XAMMP, Dreamweaver, Google Chrome and Eclipse will be used to test decision tree, website, database and e-form functionality, while SQL Inject me, and SQLmap will be used for security testing. We will also be using end user testing, and so users can be considered a resource in this case.

4.2.Testing Allocation

Sam Turner and Samuel Hickman will perform all of the black box and white box testing, which will take place between weeks 7 and 13 during the university semester. End user testing will be performed by Andrew Bryant and Jonathon Wong, as well as chosen staff from the Sydney Wildlife organisation. Testing allocation will be as follows:

Sam & Samuel: T1.1, T1.2, T1.3, T1.4, T1.5, T1.6, T1.7, T1.8, T1.9, T1.10, T1.11, T1.13, T1.14, T1.15, T2.1, T2.2

Jonathan & Andrew/Low level users: T1.5, T1.6, T1.7, T1.8, T2.1, T2.2

High level users (DB Admin, Managers): T1.12, T2.3

4.3. Testing Timeline

