**Project Specification Document**

NICTA

Ted Friedman

Eric Krenz

Sean Luthjohn

Jordan Steffan

**Table of Contents**

Section Page

[Table of Contents](#h.30j0zll)

1. Project Vision and Objectives……………………………………………………………….. 5

1.1 Project Scope and Vision 5

[1.2 Project Goals and Objectives 5](#h.2et92p0)

[2. Project Planning……………………………………………………………………………... 6](#h.tyjcwt)

[2.1 Project Lifecycle 6](#h.3dy6vkm)

[2.2 Project Setup 6](#h.1t3h5sf)

[2.3 Stakeholders 6](#h.2s8eyo1)

[2.4 Project Resources 7](#h.17dp8vu)

[2.5 Assumptions 7](#h.3rdcrjn)

[2.6 Estimates 7](#h.26in1rg)

[3. Project Tracking…………………………………………………………………………….. 8](#h.lnxbz9)

[3.1 Tracking 8](#h.35nkun2)

[3.2 Communication Plan 8](#h.1ksv4uv)

[Regularly Scheduled Meetings 8](#h.44sinio)

[Information To Be Shared Within Our Group 8](#h.2jxsxqh)

[Information To Be Provided To Other Groups……………………………........................... 9](#h.z337ya)

[Information Needed From Other Groups 9](#h.3j2qqm3)

[3.3 Deliverables 9](#h.1y810tw)

[3.4 Project Metrics 10](#h.4i7ojhp)

[4. Requirements………………………………………………………………………………. 11](#h.2xcytpi)

[4.1 Overall Description 11](#h.1ci93xb)

[4.2 Users and Roles 11](#h.3whwml4)

[4.3 Use Case Diagrams 11](#h.2bn6wsx)

[4.4 High-Level Requirements 11](#h.qsh70q)

[4.5 Low-Level Requirements 12](#h.3as4poj)

[4.6 Constraints and Limitations 13](#h.1pxezwc)

[5. Design………………………………………………………………………........................ 14](#h.49x2ik5)

[5.1 Introduction 14](#h.2p2csry)

[5.2 Scope 14](#h.147n2zr)

[5.3 High-Level Component Design 14](#h.3o7alnk)

[5.4 Activity Diagrams 14](#h.23ckvvd)

[5.5 Class Diagram 15](#h.ihv636)

[5.6 Sequence Diagram 15](#h.32hioqz)

[5.6 Data Architecture 15](#h.1hmsyys)

[5.7 Alternative Designs and Design Rationale 15](#h.41mghml)

[5.8 Low-Level Component Design 15](#h.2grqrue)

[Class 1 15](#h.vx1227)

[Description 15](#h.3fwokq0)

[Variables 15](#h.1v1yuxt)

[Methods 16](#h.4f1mdlm)

[Class 2 16](#h.2u6wntf)

[Description 16](#h.19c6y18)

[Variables 16](#h.3tbugp1)

[Methods 17](#h.28h4qwu)

[6. User Interface………………………………………………………………………………. 18](#h.nmf14n)

[6.1 UI Description 18](#h.37m2jsg)

[6.2 UI Mockup 18](#h.1mrcu09)

[7. Test Plan……………………………………………………………………………………. 19](#h.46r0co2)

[7.1 Test Plan Description 19](#h.2lwamvv)

[7.2 Testing Tools 19](#h.111kx3o)

[7.3 Test Data 19](#h.3l18frh)

[7.4 Test Types and Frequency 19](#h.206ipza)

[7.5 Test Coverage 19](#h.4k668n3)

[7.6 Test Location 20](#h.2zbgiuw)

[8. Project Closure……………………………………………………………………………... 21](#h.1egqt2p)

[8.1 Goals / Vision 21](#h.3ygebqi)

[8.2 Delivered Solution 21](#h.2dlolyb)

[8.3 Remaining Work 21](#h.sqyw64)

[9. Deliverables………………………………………………………………………………... 22](#h.3cqmetx)

[9.1 Study Results 22](#h.1rvwp1q)

[9.2 Requirements and Design Documents 22](#h.4bvk7pj)

[9.3 Code 22](#h.2r0uhxc)

[9.4 Tests and Test Results 22](#h.1664s55)

[9.5 Build Process Documents 22](#h.3q5sasy)

[9.6 Install Process Documents 22](#h.25b2l0r)

[9.7 Administrator’s and/or User’s Manual 22](#h.kgcv8k)

[9.8 Postmortem Document 22](#h.34g0dwd)

[9.9 Final Report 22](#h.1jlao46)

[10. Definitions and Acronym……………………………………………………………….. 23](#h.43ky6rz)

**1. Project Vision and Objectives**

**1.1 Project Scope and Vision**

Scope - We will format travel data provided by NICTA to be displayed the Australian national map in potentially new and creative ways.

Vision - NICTA will be able to input additional travel data and have it automatically usable by the Australian national map and have results accessible by the public in the future.

**1.2 Project Goals and Objectives**

|  |  |
| --- | --- |
| **#** | **Goal or Objective** |
| 1 | Provide full documentation on our plans for, progress on, completion of, and how to use our system. |
| 2 | Build a system which can automatically format the first set of data for the national map. |
| 3 | Make this system easy to use and presentable as a prototype in time for midterm presentations. |
| 4 | Improve the system to make it automatically format the second set of data, and all future data, to be usable for the national map |
| 5 | Come up with new and creative ways for visualization of the formatted data on the national map. |
| 6 | Present final deliverables and documentation to NICTA and for our Capstone |

**2. Project Planning**

**2.1 Project Lifecycle**

For our project our team will use an agile approach. We have met with our sponsor to gather requirements and are creating a high level development plan that accounts for them. We have three large requirements which will be broken down and outlined in our project plan. We will then follow an approach based on frequent group meetings and communication while developing a system to meet the requirements.

**2.2 Project Setup**

|  |  |
| --- | --- |
| **#** | **Decision Description** |
| 1 | Data from Australian government |
| 2 | Standards that must be followed include the default Capstone coding standard. |
| 3 | Special access privileges needed, nondisclosure forms, must be taken care of. |
| 4 | Web server for testing environment |
| 5 | Fork of National Map for development |
| 6 | Github for project documents and data used. Accessible by group, professor, sponsor, mentor, and teaching assistant, |
| 7 | Group decision to use Google Drive for collaborative work on documents. Documents will be moved to Github, when completed. |

**2.3 Stakeholders**

|  |  |
| --- | --- |
| **Stakeholder** | **Role** |
| John Grundy | Sponsor |
| Iman Avazpour | Mentor |
| Dean Knudson | Instructor |
| Sean Luthjohn | Team member |
| Jordan Steffan | Team member |
| Eric Krenz | Team member |
| Ted Friedman | Team member |

**2.4 Project Resources**

|  |  |
| --- | --- |
| **Resource** | **Resource Description** |
| Team Github Repository | Will be used to store our project documents and data. |
| Capstone Team | Our team of students who will be the primary developers of the project. |
| Iman Avazpour | The mentor who will be able to provide us with technical assistance. |
| Google Drive | Collaborative work on project documents, which when completed will be moved to Github. |
| NICTA Repository | The code and information on the National map of Australia. We will use this to try and integrate the data into the map. |
| <http://visual.bts.nsw.gov.au/lga/> | Initial project data, provided to us by our project mentor. Data will be integrated into the National Map System. |
| Cesium | Javascript api that was used to make the National map. Will be helpful in learning how the map works and the code involved in making it. |
| John Grundy | The sponsor who will be able to provide us with some guidance and help with the project as we work. |

**2.5 Assumptions**

|  |  |
| --- | --- |
| **#** | **Assumption** |
| A1 | The sample data will be continuously delivered as we need it |
| A2 | The group will be able to continuoulsy meet and make progresss |
| A3 | The group will have continuous access to computers with Microsoft Project |
| A4 | We will be allowed access to all of the neccesary data on the country |

**2.6 Estimates**

|  |  |  |
| --- | --- | --- |
| **Metric** | **Estimated Value** | **Actual Value** |
| Hours | 300 hours |  |
| Lines of Code | 1000 lines |  |
| Manual / Help Document Pages | 6 pages |  |
| Unit Tests | 75 tests |  |

**3. Project Tracking**

**3.1 Tracking**

|  |  |  |
| --- | --- | --- |
| **Information** | **Description** | **Link** |
| Code Storage | Project code will be stored on Github. | [Link](https://github.com/Fedoraman42/NICTA-NDSU-National-Map) |
| User Stories | HuBoard will be used for managing user stories and progress made on them. | [Link](https://huboard.com/Fedoraman42/NICTA-NDSU-National-Map/) |
| Continuous Integration | Will be added as necessary | N/A |
| Regression Testing | Will be added as necessary | N/A |
| Data Storage | Project data, and documents will be stored on Github | [Link](https://github.com/Fedoraman42/NICTA-NDSU-National-Map) |
| Sensitive Data Storage | Sensitive data will be stored on an encrypted Github by NICTA | N/A |
| Collaborative Work | Work on documents will be done on Google drive and then moved to Github upon completion. | [Link](https://drive.google.com/folderview?id=0B67oYcpJpbAnRUw5ajhsalRlUW8&usp=sharing) |

**3.2 Communication Plan**

**Regularly Scheduled Meetings**

|  |  |  |
| --- | --- | --- |
| **Meeting Type** | **Frequency/Schedule** | **Who Attends** |
| Conference Call | Weekly | Project team and mentor, possibly sponsor |
| Team Meeting | Twice a week & as needed | Project team |
| Short Meeting | Weekly before class | Project team |
| Sprint Planning Meeting | Start of each sprint | Project team and mentor |
| Sprint Retrospective Meeting | End of each sprint | Project team |
| Sprint Review Meeting | End of each sprint | Project team, mentor, and sponsor |

**Information To Be Shared Within Our Group**

|  |  |  |  |
| --- | --- | --- | --- |
| **Who?** | **What Information?** | **When?** | **How?** |
| Project team | Task assignments | At team meetings & as needed | Team meetings, listing in MS Project file. |
| Project team | Australian Data | When necessary as per development | Email, Github |
| Project team, mentor | Project Plan | At the beginning of each spring | Email, Skype, Github |

**Information To Be Provided To Other Groups**

|  |  |  |  |
| --- | --- | --- | --- |
| **Who?** | **What Information?** | **When?** | **How?** |
| Sponsor and mentor | Final report | At completion of project | Req./Design docs., code, Power Point presentation |
| Sponsor and mentor | Project baselines | At the end of each sprint | Onsite customer demo, access to repository |
| Sponsor and mentor | Weekly report | Weekly | Email and Trac site access |

**Information Needed From Other Groups**

|  |  |  |  |
| --- | --- | --- | --- |
| **Who?** | **What Information?** | **When?** | **How?** |
| Sponsor and mentor | Requirement changes | Start of each sprint | Conference call or meeting with sponsor and mentor. |
| Mentor | Australian Travel Data | As necessary | email, git or conference call |

**3.3 Deliverables**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Deliverable** | **Delivery Time** | |
| **Interim** | **Final** |
| 1 | Prototype Code | ▢ |  |
| 2 | Documentation on Prototype |  | ▢ |
| 3 | Prototype for Secondary Set of Data | ▢ |  |
| 4 | Documentation on Second Prototype |  | ▢ |
| 5 | Possible Adjustments to National Map for New Visualization Methods | ▢ |  |
| 6 | Final Documentation | ▢ |  |
| 7 | Finished Project | ▢ |  |

**3.4 Project Metrics**

|  |  |  |
| --- | --- | --- |
| **Metric** | **Frequency** | **Location** |
| Estimated Effort (in hours) | Per task | MS Project Plan |
| Actual Effort (in hours) | At task completion | MS Project Plan |
| Maintenance Effort (in hours) | As occurring | MS Project Plan |
| Estimated Size (in complexity points) | Per task requiring code | MS Project Plan |
| Actual Time Estimate (in Hours of work) | At completion of task requiring code | MS Project Plan |
| Meeting Times (in minutes) | Every meeting | MS Project Plan/Github |
| Weekly Meetings | Every Thursday at 5:30pm | MS Project Plan |

**4. Requirements**

**4.1 Overall Description**

<< Provide a more detailed, two to three paragraph description of the project. This description may include more technical details to describe the purpose of the project. >>

**4.2 Users and Roles**

<< Provide a list and description of the different types of users or roles within the system. This may include different classes of users, such as administrator, instructor, student, etc. This list may also include autonomous agents that interact with the system as well. These may include users (or personas) that are used as part of any user stories produced for the project. >>

|  |  |
| --- | --- |
| **User** | **Description** |
| User | A regular user who is the primary operator of the system. |
| Admin | A user who also has the ability to assign user privileges to other users. |
|  |  |
|  |  |
|  |  |

**4.3 Use Case Diagrams**

<< Provide any use-case diagrams that are being used as part of the project. Uniquely label each use case so that if necessary it is easy to reference from other parts of the document. >>

**4.4 High-Level Requirements**

<< This section lists the high-level requirements for the project and information about which of these requirements the team has committed to completed (as opposed to those which the team will not commit to, or is targeting for completion, but cannot guarantee) and when those requirements were added to the project.

The **ID** column provides a high-level ID for each requirement. This is useful for generating low-level requirement IDs which can be easily associated with a particular high-level requirement. The **Added** column lists when the requirement was added to the project as it is possible that not all requirements will be elicited or available at the project’s onset. The **Description** column gives a description of the high-level requirement. The **Status** column indicates whether these high-level requirements are committed (i.e. will be completed by the team), targeted (i.e. will be completed if the team has time after all committed requirements are completed), or not committed (i.e. will not be completed by the team).

High level requirements may also be user stories, from which multiple low-level requirements may be derived. If user stories are being used >>

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Added** | **Description** | **Status** |
| 100 | Onset | Users may view 3D representations of nodes on a globe | **C** |
| 200 | Onset | Users may view the movement of nodes over time | **C** |
| 300 | Onset | Users may view available data about a node | **C** |
| 400 | Onset | Lines of communication display relationships between nodes | **C** |
| 500 | Onset | On-click zoom functionality available to users | **C** |
| 600 | Onset | Rewind/Fast Forward/Pause/Play allows movement of nodes to be easily visualized over time | **C** |
| 700 | Onset | User interface is elegant and easy to use | **C** |
| 800 | Onset | Add halo effect around nodes | **T** |
| 900 | Onset | User may filter out different types of nodes | **C** |
| 1000 | Onset | Line segment denoting previous locations of nodes | **C** |
| 1100 | Sprint 1 | Design 3D models | **T** |
| 1200 | Sprint 2 | 3D models face the correct direction when moving | **T** |
| 1300 | Sprint 3 | 3D models are painted | **NC** |

**4.5 Low-Level Requirements**

<< This section lists low-level requirements for each of the high-level requirements. Requirements in this section should use the formal descriptive language for requirement specification (i.e. must/shall language). Additionally, some thought should be given to how these requirements will be verified and a brief description of how the requirement will be tested should be provided.

The **ID** column specifies the ID for the low-level requirement. This can be used to refer to the requirement in other documentation (e.g., the MS Project Plan) and to provide traceability. The **Description** column specifies the requirement formally. The **Verification** column provides information about how the requirement will be tested.

Remember that not all requirements are necessarily functional. Requirements can also specify requirements for security (e.g. who can and cannot use a system), globalization (e.g. descriptions and other strings will not be hard coded so that the software can be translated into other languages more easily), portability (e.g. developing an iPhone to be compatible with other smart phone platforms like Android or Windows Phone) accessibility (e.g. the program complying with ADA guidelines), availability (e.g. the system will be able to reboot in under 2 minutes in the event of failure.), etc. >>

|  |  |  |
| --- | --- | --- |
| **ID** | **Description** | **Verification** |
| 110 | Nodes shall be represented in three-dimensional space. |  |
| 120 | Users shall be able to manipulate the map to view nodes from different perspectives. | Usability test to determine if map can be manipulated in all three dimensions. |
| 130 | Node positional information shall conform to guidelines established in IEEE 702.34 guidelines. | Create test cases to verify that node positional data is stored in accordance with given standards. |
| 210 | Node information shall be updated at an interval of at least 10 minutes. |  |
| 220 | Users shall be able to determine if new node information has not been updated. | Provide indication that node information is fresh for last ten minutes. Alert user when node information is not fresh. |
| . |  |  |
| . |  |  |
| . |  |  |
| 710 | Strings for field descriptions in the user interface shall not be hard coded. |  |
| 720 | Alternative language packs shall be selectable by the user. | Create stub test data to simulate alternative languages. |

**4.6 Constraints and Limitations**

<< This section provides a list of constraints and limitations for the project. This provides additional information about any limitations that may exist in the project (e.g. will not work in versions of Internet Explorer prior to 8.0) that are not covered by requirements, but provide important information related to the project. This may include constraints related to security, performance, and other aspects of the system.

The **Constraint** field lists the constraint or limitation for the project. The **ID** field lists the related requirement ID (if any) that provides additional context for the constraint. >>

|  |  |
| --- | --- |
| **Constraint** | **ID** |
| Team will not provide alternative language pack files. | 720 |
| Provided 3D models will not be of professional-grade quality. | 1100 |
| Team will use Visual Studio 2007 for development as 2010 is not available. |  |
|  |  |
|  |  |

**5. Design**

**5.1 Introduction**

<< Briefly describe the major aspects of the design and, if applicable, how a developer will use it. For example: “Create and post a General Ledger transaction using the glTrx routines. Perform account inquiries with gjJournal routines.”

**5.2 Scope**

<< Provide a brief overview of the scope of this design. Also touch on anything that will not be included in this document. >>

**5.3 High-Level Component Design**

<< Create a diagram of the high-level components or modules in the program, linking them with arrows to show any dependencies. Also complete the tables to provide a description of each module as well as the table which traces components to their related requirements. >>

|  |  |
| --- | --- |
| **Component** | **Description** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Component** | **Related Requirements** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**5.4 Activity Diagrams**

<< Include activity diagrams for important workflows in the program. At least one diagram should be included for the main workflow in the program. Optionally include labels that indicate which component is responsible for that part of the activity. Activity diagrams for components which perform complex tasks should also be included. >>

**5.5 Class Diagram**

<< Include a class diagram for all classes to be designed. Optionally include major data elements of those classes and important methods and functions that will be used by other classes. >>

**5.6 Sequence Diagram**

<< Include sequence diagrams for important functionality of the program to indicate control flow. These diagrams should include classes found in the class diagram and use the methods for those classes to show the interaction between them. >>

**5.6 Data Architecture**

<< Include any information or diagrams that provide details about databases, xml configuration files, or other data structures that are a part of the system. If a very specific format is required, it may be worthwhile to provide a more robust description or a detailed design such as a database schema. >>

**5.7 Alternative Designs and Design Rationale**

<< Provide information regarding other designs which were considered, but not chosen. Provide rationale for why the chosen design was selected over the alternative candidate designs. >>

**5.8 Low-Level Component Design**

<< Optionally provide a detailed design for each class or module in the program. Include information about the variables, methods, and other important aspects of the class such as exception handling.

Note that the templates provided here are designed for Java and that some languages will not contain >>

**Class 1**

***Description***

|  |  |  |
| --- | --- | --- |
|  | **Access** | **Description** |
|  |  |  |
| **Extends** |  | |
| **Implements** |  | |
| **Throws** |  | |

***Variables***

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Type** | **Access** | **Description** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

***Methods***

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | **Return Type** | **Access** | **Description** |
|  |  |  |  |
| **Parameter** | **Type** | **Description** | |
|  |  |  | |
|  |  |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | **Return Type** | **Access** | **Description** |
|  |  |  |  |
| **Parameter** | **Type** | **Description** | |
|  |  |  | |
|  |  |  | |

**Class 2**

***Description***

|  |  |  |
| --- | --- | --- |
|  | **Access** | **Description** |
|  |  |  |
| **Extends** |  | |
| **Implements** |  | |
| **Throws** |  | |

***Variables***

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Type** | **Access** | **Description** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

***Methods***

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | **Return Type** | **Access** | **Description** |
|  |  |  |  |
| **Parameter** | **Type** | **Description** | |
|  |  |  | |
|  |  |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | **Return Type** | **Access** | **Description** |
|  |  |  |  |
| **Parameter** | **Type** | **Description** | |
|  |  |  | |
|  |  |  | |

**6. User Interface**

**6.1 UI Description**

<< Provide a brief description of the UI that will be used in this program and how users will interact with the program. >>

**6.2 UI Mockup**

<< Create a mockup of the user interface. This can be a simple drawing that demonstrates key parts of the user interface or a screenshot of a prototype created within an IDE. >>

**7. Test Plan**

**7.1 Test Plan Description**

<< Provide a brief description of how testing will be conducted for this project. >>

**7.2 Testing Tools**

<< Provide a brief description of any testing tools, suites, etc. that will be used to assist with testing the project’s code. >>

|  |  |
| --- | --- |
| **Tool** | **Description / Function** |
| Jenkins | Continuous integration and regression testing. |
|  |  |
|  |  |

**7.3 Test Data**

<< Identify any data that will be necessary as part of the test process. >>

|  |  |  |
| --- | --- | --- |
| **Data** | **Description** | **Link** |
| Sample Data | Sample data of known good and bad examples construct unit tests. | [Link](http://examplelink.com) |
| User Database | Access to company’s Active Directory to test authentication for users. | [Link](http://examplelink.com) |
| Order Database | Stub data for unit tests. | [Link](http://examplelink.com) |

**7.4 Test Types and Frequency**

<< Provide a list of all the different types of tests that will be performed as part of this project, a brief description of each, and the frequency with which each test will be conducted. >>

|  |  |  |
| --- | --- | --- |
| **Type** | **Description** | **Frequency** |
| Integration Test | Integration test of existing code | Weekly starting with Sprint 3 |
| Regression Test | Run unit tests against all code | Daily (2:00 AM) |
| Usability Test | Perform manual usability test of UI. | Every Sprint |

**7.5 Test Coverage**

<< Identify components or other aspects of the project that will not be tested and provide rationale for that reasoning. >>

|  |  |
| --- | --- |
| **Component** | **Rationale** |
| Req. 720 | Do not have sufficient time to generate stub data. |
|  |  |
|  |  |

**7.6 Test Location**

<< Indicate where unit tests, test cases, etc. are being stored >>

**8. Project Closure**

**8.1 Goals / Vision**

**<<** Provide an update to the vision statement that was originally stated in the Project Initiation document. >>

**8.2 Delivered Solution**

<< Provide a high-level description of what was planned and what is being delivered. >>

**8.3 Remaining Work**

<< Provide a short summary of what should be done next, ways of further improving the project, or any additional recommendations. >>

**9. Deliverables**

<< This section contains information on the location of any deliverables for the project. If there are none for a particular category, indicate that it is not applicable. Otherwise provide the name of any files, etc. and where they can be found. >>

**9.1 Study Results**

<< The location of the results of any studies performed as part of this project. >>

**9.2 Requirements and Design Documents**

<< The location of any requirements and design documents. >>

**9.3 Code**

<< The location of any code written for this project. >>

**9.4 Tests and Test Results**

<< The location of any tests (unit, regression, etc.), test results, or other testing documents. >>

**9.5 Build Process Documents**

<< The location of any documents detailing build processes. >>

**9.6 Install Process Documents**

<< The location of any documents describing installation processes. >>

**9.7 Administrator’s and/or User’s Manual**

<< The location of any manuals, or help documentation. >>

**9.8 Postmortem Document**

<< The location of the Postmortem document. >>

**9.9 Final Report**

<< The location of the Final Report document. >>

**10. Definitions and Acronyms**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| NICTA | National Information Communications Technology Australia |
| LGA | Local Government Area |
|  |  |