**MARKING SCHEDULE**

|  |  |
| --- | --- |
| Sections | Marks |
| 1. Overall | 5 |
| 2. Executive summary | 6 |
| 3. Introduction | 5 |
| 4. Data Design | 20 |
| 5. Process Design | 20 |
| 6. Architecture / Infrastructure Design | 20 |
| 7. Interface Design | 20 |
| 8. Appendices | 4 |
| Total marks | 100 |

ICT313 Neuromend

Tempest

Design Document



# Title Page



**Project name:**

Neuromend

**Client/organisation:**

Shri Rai

**Supervisor:**

Fairuz Shiratuddin

**Tempest team members:**

Ary Bizar

Anopan Kandiah

Hannah Klinac

Alex Mlodawski

Bryan Yu

**Date of document:**

20/08/2014

**Version of document:**

1.0

# Contents Page



[Title Page 3](#_Toc397597994)

[Contents Page 4](#_Toc397597995)

[Executive Summary 5](#_Toc397597996)

[Introduction 6](#_Toc397597997)

[Data Design 7](#_Toc397597998)

[Process Design 8](#_Toc397597999)

[Architecture/Infrastructure Design 9](#_Toc397598000)

[Interface Design 10](#_Toc397598001)

[Appendices 11](#_Toc397598002)

# Executive Summary



Purpose of the document is to provide a blueprint for building the system Neuromend and to provide a clear picture of the software. This document acts as a reference for the system, and will aid in the distribution of development among the development team. If the system is destroyed, it should be able to be rebuilt using this document.

This document is going to discuss data design, process design, architecture/infrastructure design, and interface design for the system.

# Introduction



The purpose of this document is to provide an outline for building the system Neuromend and to provide a clear picture of the software.

The intended audience for this document is the stakeholders associated with the project. This includes the client, supervisor and team members.

This system is the first designed and implemented product and it is called Neuromend.

Related documents:

* Requirements document
* Project management plan
* Final documentation

The above requirements document and project management plan are prerequisite documents, and provide background and context for this document and the system. Final documentation is the final document that results from this document as well as the prerequisites.

Size of the system to be implemented: 4 versions of the software simulation, each functioning with a different set of devices; Oculus Rift + Kinect, Oculus Rift + Leap Motion, Oculus Rift + Razor Hydra, Mouse + keyboard. The simulation has 3 levels. The simulation must have a networked database.

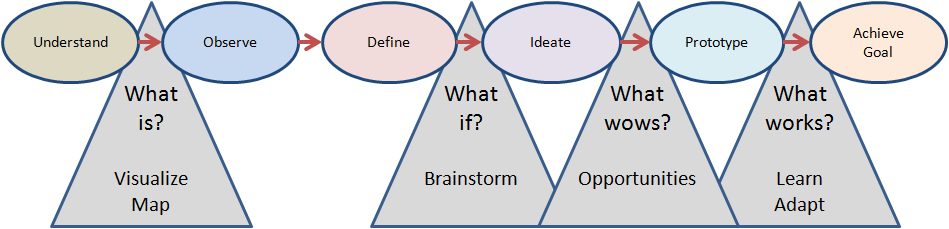
Complexity of the system: Each version of the simulation should be specific for the attached set of devices. The simulation should have a menu system. Each level should have a training stage and an execution stage. The time score for the execution stage should be stored on the user’s profile. The database is used to store user’s profiles with time scores for each level.

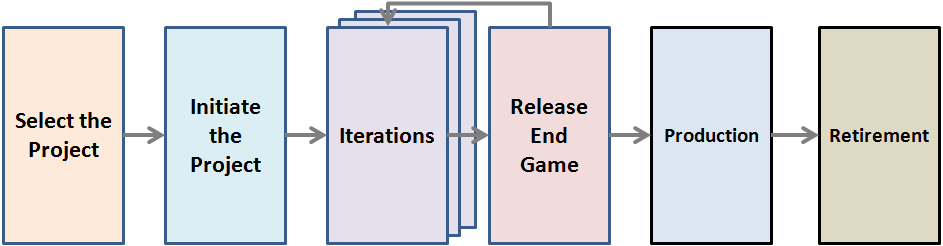
Arrangement of the design and implementation teams: 5 major areas of the system, each team member is responsible for one major area as follows:

* Level designer: Ary
* Oculus Rift + Kinect: Alex
* Oculus Rift + Leap Motion: Hannah
* Oculus Rift + Razor Hydra: Bryan
* Mouse + keyboard: Anopan

Chosen design methodology: Design Thinking. This methodology applies critical and creative thinking to understand, visualise, and develop approaches to solve the problem. Agile design is also used to break tasks into smaller increments of short time frames to deliver small working iterations to the system. The system itself has a design goal which uses design thinking, but agile design is used in smaller iterations to achieve that goal.

1. Define the problem: understand the problem, observe and understand to correctly define the right problem to be solved.
2. Create and consider options: come up with solutions to solve the problem.
3. Refine selected directions: adapt to dynamic conditions by prototyping.
4. Execute: achieve the designated goals.

Design thinking: 

Agile design:

# Data Design



Review and develop data objects, relationships, data flow and content

Bryan also done ERD

Identify all data structures and the operations performed on them

Create the data dictionary to represent the relationships among data objects and the constraints on the elements of the data structure

If using a database, perform database design

Data-to-Process CRUD Matrix if applicable

# Process Design



Detailed description of each software component:

* Process models, either traditional (DFDs & all required levels) or event-driven (decomposition diagram, event response diagrams and use case lists etc.
* Process descriptions (using structured English), expanded use-case narratives (if not already done), possibly decision tables

Address processing controls

Include algorithms, as well as an overview of the components using structure charts, hierarchy charts, etc.

If using OOM/P you will need class diagrams and sequence diagrams, either high-level with a data dictionary, or low-level which includes all the definitions.

# Architecture/Infrastructure Design



**Architecture of the system**

Neuromend is simulation software for user interaction and manipulation of objects in a virtual world. It will be installed onto PC systems and work with several sensory devices in order to achieve this. Neuromend will be a fairly small program containing just three levels for users to play through. User performance will be tracked and stored onto a sever containing a database, which will be accessible anywhere with internet connectivity. There will be four versions of Neuromend, each supporting a specific device combination.

**Architectural diagram**

The diagram below is an example of how the system’s architecture will look like.

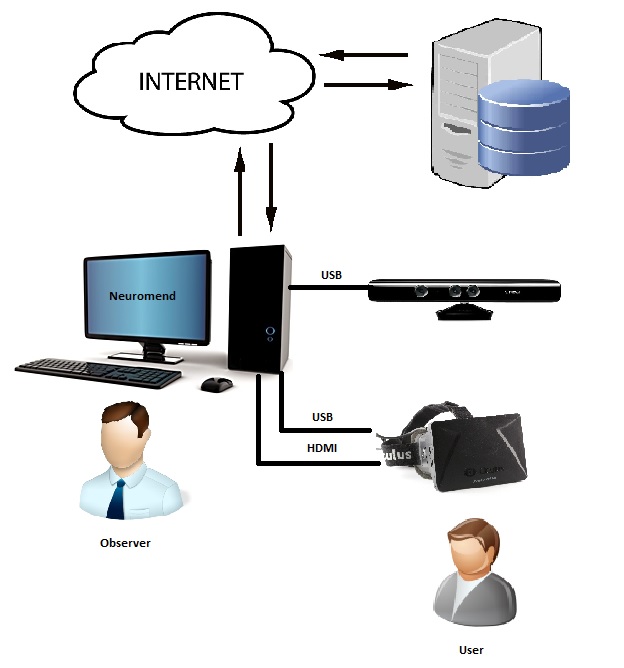


Figure 1.1 – Architecture example using Oculus Rift + Kinect combination [1].

**Additional software components**

In order for the architecture to be operational some outside software is required. These include the drivers for the sensory devices and their associated software development kits (SDK). The architecture makes heavy use of the SDKs provided with the sensory devices so they must exist on the system that will run Neuromend. All of this outside software will be included in the package.

**Infrastructure requirements:**

* **Capacity**

The system will work with one user at a time. It will have a user put on the Oculus Rift andoperate their desired sensory device or mouse and keyboard. The sensory devices include Kinect for Windows, Leap Motion and Razar Hydra. The program will have output shown on a monitor so others can observe what a user is doing. Installation of Neuromend needs to be done before operating the program so some available hard disk drive space will be required. Transactions will occur between the system running Neuromend and a server so a stable internet connection will also be required. These transactions involve sending and receiving users’ performance data.

* **Performance**

Systems that will run Neuromend will need to be relatively modern in order to smoothly display its graphics. This is important because slow/jerkiness will be disorientating for users, which will negatively impact their performances because their speed of each level completion is timed.

* **Integration & compatibility**

As mentioned in the performance section the system that will run Neuromend will need to be relatively modern. It will require a decent graphics card, internet connectivity, available hard disk space and USB ports. Compatibility between the sensory devices is required for operation. Systems will need compatibility with the following sensory device combinations Oculus Rift + Razer Hydra, Oculus Rift + Leap Motion, Oculus Rift + Kinect or a keyboard and mouse.

* **Platform strategy**

Systems will need to have Windows 7 to run Neuromend. Support and compatibility cannot be guaranteed on other versions of Windows or Macs because of the developmental nature of the sensory devices.

* **Security**

Neuromend will require users to input some personal data about themselves in order to accurately track their performances. This data will be stored on the system running Neuromend and a server communicated with through an internet connection. To make such data unidentifiable if there ever is a breach users’ will be identified using a unique ID instead of by names. Data stored in the database will be protected and have restricted access to those who have the right permissions. Access will be done through a log in system as a form of authentication.

* **Back-up & recovery**

Local failure on a system running Neuromend will not be a big issue because user data will still be stored on a sever so a recovery can be performed by getting lost user data back onto the system once it’s running again. However, there is a chance that the server may fail so local backs-up should continually be made by users. This should involve copying data over to external storage in case both their system and the server fail at the same time.

* **Scalability**

Design of Neuromend focuses on separation of functionality so it can be scaled up easily if required. This may include a multiplayer option so more than one user can play Neuromend at a time. It may also include using additional devices for finer control.

* **Future proofing**

The sensory devices developmental nature means that they will likely change quickly with time. Neuromend is not designed not to force users to upgrade devices should new ones become available. Updating of sensory devices may require patches for Neuromend to work with any new SDKs that accompany such devices.

**Alternative designs**

An alternative of designing four separate versions of Neuromend is to design just a single one that contains options in the software itself to switch between device combinations. However, this would prove fairly difficult because it was discovered early on that some of the sensory device drivers’ conflict with each other. This could cause system failures or device inoperability.

# Interface Design



The design of the interfaces between software modules:

* The design of interfaces between software and non-human (external) entities.
* The design of the interface between human and computer (HCI)

You should provide mock screen-shots of the interfaces of the system you intend to create.

Interface specifications: input/output controls, formatting, etc.

# References

<http://www.nwlink.com/~donclark/design/design_models.html>



<http://www.rockwellautomation.com/resources/images/rockwellautomation/services/Generic_Computer--graphic_280w.jpg>

<http://www.dallmeier.ru/fileadmin/upload_electronic/Unternehmen/Niederlassungen/Dallmeier_Russland/Planning_icons_single/Shapes/JPG/Symbols/Cloud%20internet.jpg>

<http://www.advess.net/media/database_server1.jpg>

<http://www.blograzzi.net/wp-content/uploads/2011/04/arrow.jpg?234d14>

<http://www.tedesys.es/sites/default/files/kinect.jpg>

<http://upload.wikimedia.org/wikipedia/commons/a/ae/Oculus_Rift_-_Developer_Version_-_Front.jpg>

<http://www.softicons.com/social-media-icons/free-social-media-icons-by-aha-soft/user-icon>

<http://www.softicons.com/web-icons/user-icons-by-2shi/user-1-icon>

# Appendices



Appendix A: Deliverable Task Breakdown Statement

Appendix A:

