## Requirements Analysis Document

**Group F  
Revision History:**

Version 1 Created – 22nd August, 2011  
  
**Preface:**  
This document is designed to outline the expectations of Kwik Browser rewrite for the developers and client.

**Target Audience:**

Client, Developers

**Members:**

Thomas Bruining

Haydn Muir

David Kucsai

David Turnbull  
Basimah Aljhne

Abdulraham Qasem

**MILESTONES**

* 22nd August – Created Initial R.A.D.

**1.0 General Goals**

**Purpose of the System:**To provide a cross platform, easy to use, quick and easy to learn Web Browser based on the existing Kwik Browser implementation.

**Scope of the system:**Basic web browser and navigation  
Cross platform at least for Windows and OS X.  
Search bar built in  
History  
Favourites  
Self contained (No installer)  
  
In-built text based dynamic advertising (The functionality for advertising will be built, we will not build an advertisement serving platform, though will provide documentation on how it functions/the way in which to structure the information)  
Popup Blocker

Tabbed browsing  
Keyboard Shortcuts

**2.0 Current System**

We have a functioning web browser written in Visual Basic. However the current source code is unlikely to facilitate the building of the new browser in a major way.

**3.0 Proposed System**  
  
  
**3.1 Overview**  
We propose to build a web browser using SWT, Java and rendering via the Operating System’s default web browser engine.

**3.2 Functional Requirements**Browses the internet.  
Javascript functionality.  
History  
Favourites  
Navigation  
Search Bar  
Cookies  
Security (Pop up blocker)  
HTTPS support  
  
  
**3.3 Nonfunctional Requirements**Interface redesign  
Fast startup, fast interface changes/switches/feedback  
Portability – No install, self contained.

**3.3.1 User Interface and Human Factors**

Novice users/Target Audience  
Obvious feedback  
simple to understand tooltips/descriptions  
Static toolbars

Minimal menus/options  
  
**3.3.2 Documentation**

How to/tutorial document on changing dynamic advertising  
How to manual for use of browser  
Acceptance Tests  
Project Plan  
  
**3.3.3 Hardware Consideration**

Slow/Old Hardware

Recommended to be running current java version  
Low resolution screens – Working on the lowest of 800x600  
Assumption that default browser already exists  
Minimum operating system Windows XP and Mac OS X

**3.3.4 Performance Characteristics**

Start up time on par with most common Internet Browsers.  
 Average startup time over 10 tests within at least 30% of Kwik Browser start up time.  
The speed of web page rendering will be limited by the Operating System’s default web browser and the users internet connection.  
 We can not optimise the web page renderer as it is not within our scope.

**3.3.5 Error Handling and Extreme Conditions**

“This page cannot be found” – The loaded page will have a link redirecting them to their chosen search engine with the user inputted URL as it’s search term.

No network connection – The user will be alerted to the fact that their network connection is not connected.

Automatically correct user input in URL bar for simple issues such as missing http:// or www.

**3.3.6 System Interfacing**

The system will interface with the operating systems default web browser.

It is expected that the user will have typical input devices (Mouse, keyboard)

**3.3.7 Quality Issues**

The browser must be portable and self contained. (No installer)

The system should be capable of supporting secure websites.

**3.3.8 System Modifications**

Dynamic text based advertising

History – Clearable

Favourites – User added

Set home page

Set search engine

Popup Blocker On/Off

**3.3.9 Physical Environment**

Refer to system requirements.

**3.3.10 Security Issues**

Must support HTTPS and have a functioning popup blocker.

The system will not save passwords.

**3.3.11 Resource Issues**

The browser should be small enough to download quickly from a webserver with a standard home interenet connection

**3.4 Constraints**

We have decided that Java will be the language we use to build the system using a set of libraries freely available to the public.

**3.5 System Model**

INSERT DIAGRAM HERE

**3.5.1 Scenarios**

For this section, think about all the possible ways which the users will interact with your subsystem. Present them in a "story" format.

**3.5.2 Use Case Models**

TO BE ADDED

**3.5.3 Object Models**

INSERT HERE

**3.5.4 Dynamic Models**

**3.5.5 User Interface - Navigational Paths and Screen Mockups**

