2011

Requirements Analysis Document (RAD)

Kwik Browser

CITS3200 - Professional Computing Second Semester 2011 University of Western Australia Perth, WA 6009

UWA G:F
8/26/2011

Revision History (Document Change Control)

Version	Date	Description	Author
0.1	22nd August, 2011	Created the Document from the template provided	
0.2			

Table of contents

Tables and Diagrams

Preface:

This document is designed to outline the expectations of Kwik Browser rewrite for the developers and client.

Target Audience:

Client, Developers and project supervisor.

Members:

NO.	Name	Student #	Role	Organization
1	Gerald Veurink	N/A	Stakeholder/client	
2	Thomas Bruining	20491947	Project manger	UWA
3	Haydn Muir	20500093	developer	UWA
4	David Kucsai	20495791	developer	UWA
5	David Turnbull	20496608	developer	UWA
6	BasimahAljhne	20475731	developer	UWA
7	AbdulrahamQasem	20473069	developer	UWA

Table 1: project member

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:

- Must:
    1. Basic web browser and navigation.
    2. Cross platform at least for Windows and OS X.
    3. Search bar built in.
    4. History and Favorites.
    5. Self contained (No installer).
- Should:
    1- In-built text based dynamic advertising (The functionality for advertising will be built, we will not build the advertisement serving platform).
    2- Popup Blocker.
- Could:

1- Tabbed browsing.

2- 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

Priorities	Name	Description	Client Value	Estimated Hours	Estimated Time Rat
1	Basic web browser navigation	Forward, back, stop, home, etc.	$15	20	0.75
1	Windows and Mac OSX Support	Support on the two major	$15	6	2.5
2	Search bar built-in	A search bar with various selectable engines	$10	5	2
2	Self contained	No Installer Files are contained in one area	$10	10	1
2	Popup Blocker	Block unwanted popups when turned on	$10	10	1
2	Tabbed browsing	Multiple Tabs in one Window	$10	15	0.66
3	Keyboard Shortcuts	Home, back, etc.	$10	3	3.33
3	In-built text based advertising	Text Based advertising pulled from an external server.	$5	10	0.5
4	Favorites		$5	10	0.5
4	History		$5	10	0.5
TOTAL	NA	NA	$100	105 Hours	

3.2.1 Value Estimate Ratio Ranking

Ranking	Feature Name	Estimated Value Ratio
1	Keyboard Shortcuts	3.33
2	Windows and Mac Support	2.5
3	Search Bar Built-in	2
4	Self Contained	1
5	Popup Blocker	1
6	Basic Web-browser Navi	0.75
7	Tabbed Browsing	0.66
8	In-built text based advertsing	0.5
9	Favorites	0.5
10	History	0.5

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

1- Novice users/Target Audience
2- Obvious feedback
3- Simple to understand tooltips/descriptions
4- Static toolbars
5- 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

3.3.2 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

3.3.3 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.

3.3.4 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 active.

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

3.3.5 System Interfacing

The system will interface with the operating system's default web browser.

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

3.3.6 Quality Issues

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

The system should be capable of supporting secure websites.

3.3.7 System Modifications

4 Dynamic text based advertising
5 History – Clearable
6 Favorites – User added
7 Set home page
8 Set search engine
9 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 internet 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

Nor provided on this version of the document.

3.5.1 Scenarios

This scenario addresses the positive aspects of how a tabbed browsing is a powerful addition.

A user opens the Kwik Browser and sees their home page – Yahoo.com. The user want to check the personalized news headlines on the home page. They then scroll down to check their local weather forecast and open that in new tab and clicks on the Finance link which open in another new tab. He read the first loaded tab by clicking on it and check the details of the weather. When he done with that, close it and read the other opened tab. So he will never have to had wait for anything to load.

To highlight the features that make Kwik Browsa quick to use.

George has a busy daily schedule but likes to check the news each morning using Kwik Browsa, in between getting ready for work. He turns on his computer, clicks the Kwik Browsa executable and it launches immediately. He begins to type in the website of his favourite news website but remembers it will be in his history so clicks the history button and selects it from the list. The page loads quickly and he is able to catch up on the latest headline. Once done he decides to add the website to his favourites to save even more time when he next wants to check the news so he clicks the favourites button and adds it to his favourites list. He then closes the Kwik Browsa window and continues on with his daily routine.

Focussing on the history and favourites:

Fred decides to check some websites one day. He opens Kwik Browsa and types in a website url in the address bar. After browsing this website for a while, he remembers a website he found useful the previous day. He then opens the history tab and finds the website he was looking for and clicks it. The browser then loads the website. He then browses the website a bit longer then decides he may like to come back to this website again in the future. Fred opens the favourites tab, he then, clicks the add favourite button. Kwik Browsa then stores the website in the favourites for later access. Fred finishes his browsing and so closes the Kwik Browsa.

Focussing on Portability, simple to use interface and our target audience.

Jeff gives free lessons for people who are new to computers at his local community centre. In this particular lesson he wants to teach the class of (mainly) elderly people how to access the internet. Before the lesson he puts the browser on each computer's desktop, it doesn't require him to install anything to the system. He just makes a copy of the program from his USB stick. He finds this very convenient and portable; a machine just needs to have Java installed.

He starts up Kwik Browsa on his computer just to make sure everything is in working order. The browser displays a page telling him that a network connection isn't available. He makes sure that he is connected to the network and then refreshes the page. It loads up his homepage. Everything is in working order.

During the class Jeff is very happy because the users are finding it very easy to interact with the program and get it to do what they want it to do. This is due to the interface being designed to be easy to look at and understand. At the previous class Jeff instructed the class to get USBs if they can so he can distribute Kwik Browsa for them to use at home, otherwise they can download it off the web. Jeff considers Kwik Browsa to be a great tool for teaching people the basics of browsing the internet and once they become more confident with it they can start performing more advanced actions with it such as using shortcuts to browse and saving favourite webpages.

3.5.2 Use Case Models

Deliverable B

3.5.3 Object Models

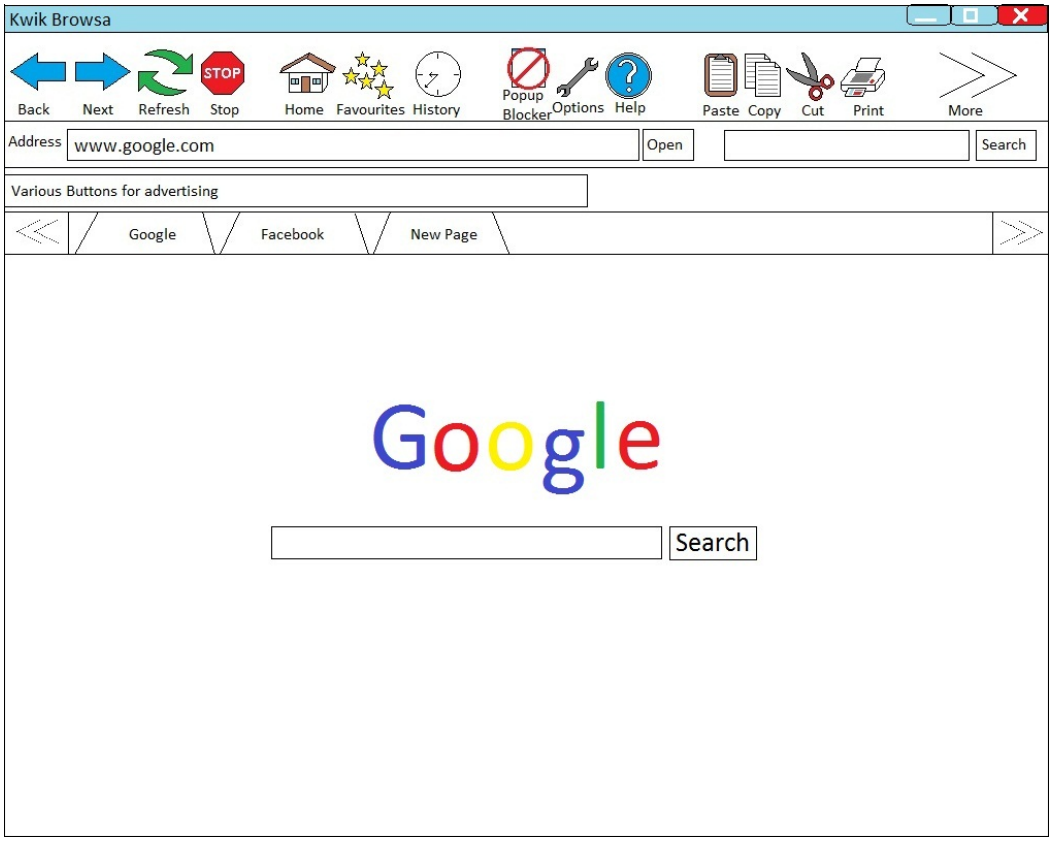
Deliverable B

3.5.4 Dynamic Models

Deliverable B

3.5.5 User Interface - Navigational Paths and Screen Mockups

3.5.5 User Interface - Navigational Paths and Screen Mockups



Notes on how the UI works:

- Favourites and history will have small pop out sections to save, load and modify as required for each. (as shown below)

- The ‘more’ button on the far right of the icons will only appear when a window has been made smaller to provide access to buttons that would otherwise appear off screen.

- The ‘New Page’ tab will always be available and all a user has to do to open a new tab is to select this page and start operating as normal.

- ‘Back’ and ‘Forward’ buttons grey out when the function is not available (eg. The user is currently navigated to the furthest forward page visited)

This is the page for the favourites section. It will appear just under the 'favourites' button when the button is pressed. The history page will work and appear in a similar fashion.

4. Glossary

Browser rendering engine – The backbone of web browsers, in charge of controlling how a page displays taking into consideration all the rules and conventions of various online protocols and languages.

Javascript – A scripting language designed for the web.

Web Browser – A program which is used to navigate the internet. It generally features navigation tools and can have additional features built into it.

Favorites – A list of websites a user manually selects which can be used to access a website quickly.

History – A record of previous websites accessed by the user.

Popup – An unwanted window created by a website, usually advertising.

Java – A system produced by Sun to power java code on people's computers. Usually installed on Macs, sometimes requiring a download on Windows PC's.