# Appendix

# A. Software Application Design

## A.1 Application Architecture

The application is designed according to a three-tier architecture model. The model consists of different clients which communicate with the application. The client interacts with the presentation layer of the application. The application controls the business and data layers.

Client

presentation layer

presentation layer

Client

Application

Business layer

Data layer

Figure 1: Application three-tier architecture

## A.2 Software Iterations

Application functionalities are defined into two distinct iterations.

Annotated and topographically designed web page

Definition of voice commands

Browser navigation using voice commands

Figure 2: Iteration 1

Improve usability and flexibility of voice commands

Improve navigation performance

Improve visual annotations of the web page

Figure 3: Iteration 2

## A.3 Web Page Layout Designs

The outer textbox represents the web page or GUI. The individual boxes represent the different components of the web page and illustrates how these boxes are arranged for each layout style.

Web page

Forward 3

Back 2

Home 1

Image1 5

Link1 4

Link2 6

Link3 7

Image3 9

Image2 8

Figure 4: Box web page design

Web page

Back 2

Home 1

Forward 3

Link1 4

Image1 5

Link2 6

Figure 5: Hierarchical web page structure

Web page

Image1 5

Link1 4

Forward 3

Home 1

Back 2

Link2 6

Figure 6: Carousel-styled web page layout

The boxes on the outer regions of the web page are visible as the slide show is navigated through.

# B. Project Background

## B.1 Functionality Requirements

The application shall:

* Be designed for a laptop/desktop computer.
* Be screen and voice oriented.
* Be deployed as a web page using a local host.
* Be created as an add-on or extension to an existing browser.
* Be designed for compatibility across browsers with a cross-platform web page interface.

## B.2 Design Constraints and Assumptions

The following constraints are presumed:

* Computer illiterate elderly users will require training.
* A time limitation of 8 weeks is defined.
* Cost is a considerable factor which shall be minimised through the use of Open Source Software (OSS).

## B.3 Quality Attributes

The quality attribute requirements are determined as:

* Buildability: The application shall be built in 8 weeks.
* Usability. The application shall be easy to use and interact with. Layout and style of the application will be user friendly. Operations shall be cancelled within one second.
* Performance. The application shall respond within one second to defined voice commands. Users will be provided with adequate feedback during the execution of the application.
* Reliability. The application will exhibit 100% reliability, i.e. no – run time or overflow – errors thereby causing the system to crash.
* Flexibility. The application shall be designed for suitability amongst elderly users over the age of 50 and with 100% adaptability across different versions of the Windows operating systems.
* Maintainability. Patches and upgrades shall be implemented in a practical manner for easy integration into the original application within 1 hour.

# C. Developer Tools

## C.1 Operating System

There are two major classes of operating systems: Windows and Unix. The Windows operating system (OS) provides a GUI OS which allows the user to interact with the computer system through mouse and keyboard inputs. Unix operating systems are interacted with by using command-line applications or GUIs. Features of each operating system are listed [19].

Unix:

* Released a Linux OS which is designed for micro-computers. Linux is freely available and an inexpensive option for most developers.
* Possesses greater processing power than Windows.
* Is flexible and can be installed on different types of machines such as supercomputers and main-frame computers.
* Software upgrades are freely available.
* Offers a range of inexpensive operating systems that present developers with flexibility and control in the design of the operating system.
* Is a leader in web serving as many internet based systems rely on Unix operating systems that run Apache.

Windows:

* Became popular due to the increasing power of micro-computers with Intel-compatible processors.
* Microsoft’s marketing strategy ensures Windows operating systems are already installed into PCs when they are purchased. As a result many consumers have no idea of what an operating system is and the different types that are available.
* Software upgrades must be purchased from Microsoft but are comprehensive and extensible.

A comparison between the two operating systems results in Windows being the optimal operating system. Unix offers immense processing power and is a leading web server. However the browser application will be rendered and does not require hosting and other web resources offered by Unix. An existing operating system is required as the application requires no changes to the underlying operating system of the machines. The browser application is a standard application that requires minimal functionality from the operating system. This supports the argument that most users are unaware of the current operating system in their PC and other types of operating systems. A major determining factor is that the application is designed for elderly users and hence Windows is the most suitable OS.

## C.2 Programming Languages

The following web programming languages are considered [20]:

* Java is a leading full-featured programming language targeted at web applications. The advantages of Java include platform independence, high performance and safety. However many developers feel that Java is not as simple as suggested and is merely a cleaned up version of C++.
* JavaScript is a scripting language which is used to integrate HTML and Java applets. JavaScript differs from Java in that byte code is interpreted and not compiled. JavaScript is a weakly typed scripting language aimed at extending the capabilities of HTML whilst maintaining the usability of HTML. JavaScript is limited as it is much more complex than HTML and is not accessible to users as HTML.
* Python is an object-oriented scripting language. Python is interpreted and resulting Python applications are cross platform. Python’s syntax evokes C and C++ and is considered a clear and intuitive programming language. Python offers a portable GUI library although certain developers suggest that Python does not support regular expression matching and native output formatting.
* Visual Basic Script (VBScript) is Microsoft’s defined internet scripting language. VBScript source code is embedded in HTML and can be downloaded to the client in an HTML file. However VBScript has no GUI development abilities and is a subset of Visual Basic which is very low level programming language [20].
* The Hypertext Preprocessor (PHP) programming language is a widely-used scripting language suited for the production of dynamic and interactive web pages. PHP is a server-side scripting language which can be embedded in HTML, i.e. the output from PHP code is combined with HTML in a script and the result is sent to the user as a web page [21].
* Ruby is an interpreted object oriented scripting language. Variables in Ruby are not types and the language syntax is simple and consistent. Memory management in Ruby is automatic [22]. C extensions can easily be written in Ruby using an API which calls Ruby from C. These calls include methods for embedding Ruby in software and for using Ruby as a scripting language [23].

Ruby On Rails is a web application development framework written in Ruby. Ruby on Rails makes assumptions about what developers require to get started and therefore makes programming web applications easier, i.e. the Rails philosophy supports convention over configuration [24].

Although Python is an object oriented language Python does not support regular expression matching and local output formatting. Microsoft’s VBScript does not provide GUI development functionalities. PHP is a server side scripting language and the browser application is focused on client side rendering. Hence PHP is not suitable. Although Java is a resourceful platform independent language well suited for web development, a scripting language would be more suitable to the design of the application. Scripting languages make direct calls to the operating system and provide immediate feedback during development. Scripting languages support powerful string operations and regular expressions [23]. Subsequently JavaScript and Ruby are applicable scripting programming languages to consider for the development of the browser application.

HTML is used to format web pages. HTML communicates with a browser and instructs the browser on when to start and end paragraphs, the types of headings to use, when to use a list to display items and when forms are required [25].

## C.3 Programming Environment

Aptana Studio is a leading open-source web application IDE that supports quick and easy development of web applications. Aptana exploits the flexibility offered by Eclipse and can be used for building, editing and debugging of both JavaScript websites and Ruby on Rails web applications [26].

The Cloud9 IDE supports several languages including JavaScript and Ruby. Cloud9 is an IDE that runs in the browser and exists in the cloud, i.e. the IDE is built on standard HTML, JavaScript and CSS. This allows the developer to run, debug and deploy applications from anywhere and at any point in time. Cloud9 offers advanced editing features and real time code analysis which reduces time spent on additional debugging. Cloud9 provides developers with a unique application URL for test and debug purposes [27].

Ruby Development Environment (RDE) is strictly a Ruby IDE and editor for Windows. RDE allows developers to edit, run and debug scripts from within the IDE. RDE provides developers with the opportunity to use Ruby as a GUI application as Ruby is actually a console application. RDE can be used as an editor to write, debug and execute scripts [28].

JavaScript Development Tools (JDST) is strictly a JavaScript IDE for Eclipse [29]. Eclipse is an open source community focused on creating an extensible development platform and various application frameworks for deploying and managing software [30].JSDT is based on Java Development Tools (JDT) and provides advanced editing features, smart error detection and correction. JDST is a fast and extensible framework designed for JavaScript development needs [29].

## C.4 Version Control

Mercurial is a free, distributed version control management tool. Mercurial provides efficient management of large projects and offers a wide range of functionalities which are enhanced with the use of shell extensions [31]. BitBucket is a free code hosting site for Mercurial. BitBucket allows team members to locally commit to the repository independent of a network connection. BitBucket assists in monitoring and managing the activity of the project source code [32]. TortoiseHg is a Windows shell extension which can be used as a GUI to Mercurial. The extension provides basic and advanced control system functionalities. TortoiseHg is simple to use to commit and retrieve source files from the repository and is efficient in instances where team members work concurrently on the same source code. TortoiseHg provides a basic user friendly GUI to aid developers in revision control [33].

Git is an open source distributed version control system. Git is a quick and efficient repository which provides extensive revision tracking capabilities. This revision management system provides version control of files similar to that of Mercurial and in addition is not dependent on network connectivity or a central server [34]. TortoiseGit is a Windows extension which provide a GUI to Git. TortoiseGit provides an easy way to commit files, show the project history log and create branches and tags [35].

Mercurial and Git are both freely available version control systems. The main difference between the systems is in the experience of using the system, i.e. how the system feels during use. Version control systems are used daily and the feel of the system is an immense deciding factor. Mercurial offers a modular control philosophy with small and efficient commands. Whereas each Git command provides a range of functionalities which are an efficient way of controlling files. Mercurial uses BitBucket as a code hosting site and Git uses GitHub. GitHub offers many features above BitBucket such as graphical design and speed [36].

Both Mercurial and GitHub use Tortoise shell extensions which provide a GUI to the version control system. Both version control systems are offer similar functionality but differ in style and advanced features. Git appears to offer more resources and functionality to assist developers.

# D. Project Plan

## D.1Process Model

### D.1.1. Agile Methods

The following agile methods are considered:

* Incremental Development

In this development model the functionality of the application is produced and delivered to the customer in increments also known as steps. In each step the requirements obtained are compared against those that are required. Incremental development focuses on the essential features first and additional functionality is only included if possible and necessary. In incremental development the user is directly involved in planning the next step and as a result changes can be incorporated more easily into each step [18,60-61].

* Rapid Application Development (RAD)

RAD is an iterative development model which focuses on user participation in the application design. RAD implements a time box element which consists of a fixed time frame within which specified functionalities must be completed. In the event certain functionalities are not completed, they are simply discarded. The RAD lifecycle consists of four phases: requirements planning, application design, construction and cutover. In addition RAD employs a triage process which prioritises requirements according to the categories denoted by MoSCoW: must haves, should haves, could haves and won’t haves. Must haves are requirements that are definitely needed whereas should haves are important requirements that are not essential to the usability of the system. If time is available, could haves are implemented and won’t haves are requirements that are omitted for the next iteration [18,62-63].

### D.1.2. RAD Process Model

Four phases of the application life cycle are defined: requirements, design, implementation and delivery. The requirements phase involves researching the various components of the project and defining the features and behaviours required by the application. The design phase develops a suitable model of the application in terms of the structure and components of the application. The implementation phase is the actual coding of the application and the analysis phase evaluates the application to determine if all requirements are met. If certain aspects are not implemented and there is no longer time available these aspects are discarded.

The requirements stated are prioritised and additional requirements are considered and categorised according to the MoSCoW categories. Timelines for the implementation of the requirements are indicated in appendix D.2.

Must haves:

* The application shall be designed for a laptop/desktop computer.
* The application will compose of defined voice commands and a clearly annotated web page.
* Integration between voice commands and annotation references shall be indistinguishable.
* Different numerically annotated web page structures shall be compared.
* The user interface is screen and voice oriented.
* A local host shall be used to deploy the application as a web page.

Should haves:

* Intuitive and smooth navigation.
* A cross-platform web page interface will be designed for compatibility across browsers.

Could haves:

* The application responds to users both graphical and vocally. The application interacts with the users by returning voice commands to the user in a pre-recorded voice.
* Users are able to choose between a male or female voice.

Won’t haves:

* Improve visual annotations of the web page.
* Improve usability and flexibility of voice commands.
* Improve speed and navigation performance of the browser.

## D.2 Project Organization

Table 1: Time management of application development process

|  |  |  |  |
| --- | --- | --- | --- |
| **Tasks** | **Start Date** | **Duration (days)** | **End Date** |
| Individual components | 2011/09/05 | 21 | 2011/09/26 |
| Integration of components | 2011/09/26 | 7 | 2011/10/03 |
| First iteration testing | 2011/10/03 | 7 | 2011/10/10 |
| Start documentation | 2011/10/10 | 7 | 2011/10/17 |
| Second Iteration | 2011/10/10 | 7 | 2011/10/17 |
| Testing second Iteration | 2011/10/17 | 3 | 2011/10/20 |
| Finish documentation | 2011/10/20 | 8 | 2011/10/28 |

The first iteration consists of the implementation of the individual components assigned to team members and the integration of these components. The first iteration is expected to be completed on 2011/10/03. Upon comprehensive testing of the iteration the final report will be started. Thereafter on 2011/10/10 the second iteration will commence and due to the small implementations included in the second iteration, testing of the second iteration is expected to take a total of 3 days. The remaining time will used to complete the documentation.

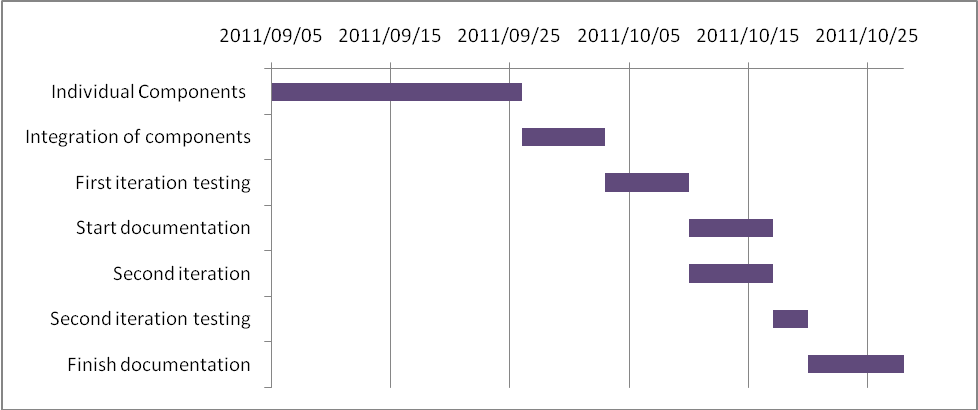


Figure 7: Gantt chart depicting project milestones

## D.3 Standards, Guidelines and Procedures

Configuration management software will be used to control – source code, third party libraries, APIs, environment settings and additional software components – amongst developers. The above components will be added to the configuration system which will ensure correct procedures are followed for version control, defect tracking and software releases. Documentation can also be stored on the system and is accessible to developers.

The application will be designed according to the International Organisation for Standardization (ISO) 9000 standard. ISO 9000 consists of a series of standards for quality management in systems [37].

The Association of Computer Machinery (ACM) Code of Ethics and Professional Conduct will be abided by [38]. The Institute of Electrical and Electronics Engineers (IEEE) Code of Ethics will be adhered to during and after project execution [39].

## D.4 Risks

Due to the time limitation of 8 weeks the following risks are identified and corresponding measures are suggested to deal with the risks:

* If an add-on for an existing browser cannot be created within three weeks, a new browser will be designed and created for the application.
* In the event defined voice commands cannot be utilised and accessed alternative voice methods shall be considered.
* Provided the web page can be adequately numerically annotated, if time is limited only one layout style will be implemented.
* If only the first iteration is implemented during the project timeline, the product output of the first iteration will be delivered.

## D.5 Methods and techniques

There are three phases in the project lifecycle: design, implementation and testing. During the design phase aspects such as abstraction, modularity, information hiding, complexity and system structure are considered. These aspects affect the maintainability and reuse of the software [18,330]. During the implementation phase the components will be implemented and sequentially tested by the developers.

Possible test cases are proposed in Section 7. The individual assigned voice and GUI components will be tested using unit testing methods. Thereafter the components will be integrated into the browser navigation application and integration testing will be performed to evaluate the composition of components into the application. Elderly users will be educated on how to use the software application. Thereafter a series of tests will be performed for acceptance testing to determine the usability of the application [18,449].

## D.6 Work Package

### D.6.1 Work Allocation

The application is divided into three components: voice, user interface and browser navigation. These components are equally distributed between the two team members. The component functionalities are shown in table 2. The iteration in which each functionality will be implemented is indicated.

Table 2: Work allocation between team members

|  |  |  |  |
| --- | --- | --- | --- |
| **Team Member** | **Component** | **Functionality** | **Iteration number** |
| Kirti | GUI | Annotated and topographically designed web page | 1 |
| Improve visual annotations of the web page | 2 |
| Cole | Voice | Definition of voice commands | 1 |
| Improve usability and flexibility of voice commands | 2 |
| Kirti, Cole | Browser navigation | Navigation using voice commands | 1 |
| Improve navigation performance | 2 |

### D.6.2 Individual Work Schedule

The web page component is further decomposed into tasks. The tasks are prioritised according to the level of functionality involved. Corresponding daily time frames are applied to impose a limit on the completion of each task as shown in table 3. These tasks belong to the first iteration and correspond to the time management shown in the appendix D.2.

Table 3: Time management for web page tasks

|  |  |  |  |
| --- | --- | --- | --- |
| **Tasks** | **Timeframe (days)** | **Start Date** | **End Date** |
| Creation and deployment of the web page | 4 | 2011/09/05 | 2011/09/08 |
| Layout and structure | 4 | 2011/09/09 | 2011/09/12 |
| Graphical images and text | 3 | 2011/09/13 | 2011/09/15 |
| Numerically annotated links | 4 | 2011/09/16 | 2011/09/19 |
| Different GUI layout styles | 4 | 2011/09/20 | 2011/09/24 |
| Integration with voice commands | 4 | 2011/09/26 | 2011/09/29 |
| Browser navigation using voice commands | 4 | 2011/09/30 | 2011/10/03 |

The tasks defined in the second iteration consist of the improvements of the visual annotations of the web page and the performance of the navigation and will be implemented in the week of 10 – 17 October 2011.

### D.6.3 Team Work

Team members are assigned respective components and project milestones are defined. In addition each team member has created an individualised work schedule. A configuration management system will be used to control the source code and project files of both team members. To ensure members are working consistently members will communicate via electronic mail (email) and instant messaging services such as BlackBerry Messenger (bbm) and Goolge Talk (gtalk). All queries will be addressed immediately and changes to the project will be assessed for feasibility and accordingly appended to the project timeline.

## D.7 Resources

Open Source Software and Freeware products will be utilised in the project to minimise cost factors. These products are freely available on the internet and many have limited free trial periods before a license must be purchased to continue using the product. Desktop/Laptop computers will be used to implement the application and are available for use at the University. Internet connectivity will be required for research and deployment purposes and is provided by the university.

## D.8 Delivery

The application will be locally deployed onto desktop/laptop machines for users. The application will be delivered as an add-on to existing web browsers such as Mozilla Firefox and Microsoft’s IE. However for testing purposes the add-on will be appended to the browser before the testing phase commences.

# E. Testing

## E.1 Comparative Testing

Table 4: Assessment of numerically annotated references

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Assessment Criteria** | **Scale** | | | | | |
| **Poor** | **Acceptable** | | **Good** | | **Excellent** |
| Usability |  |  | |  | |  |
| Performance |  |  | |  | |  |
| Reliability |  |  | |  | |  |
| Flexibility |  |  | |  | |  |
|  | | | | | | |
| **Time Assessment** | **Slow** | | **Acceptable** | | **Quick** | |
| Time |  | |  | |  | |