**ENG 3051 – Software Engineering Design II**

**Fall 2015**

**Stereo Camera 3D Reconstruction**

**Contributing Members:**

**Slim Babay, Antoine Stavro**

**Date Submitted: 10/2/2015**

Approval

This document has been read and approved by the following team members responsible for its implementation:

|  |  |  |
| --- | --- | --- |
| **Print Name** | **Signature** | **Comments** |
| Slim Babay |  |  |

|  |  |  |
| --- | --- | --- |
| **Print Name** | **Signature** | **Comments** |
| Antoine Stavro |  |  |

**Revision Page**

**Table of Content**

# Statement of Problem & Project Objectives

### Statement of Problem:

In a large database of 150+ resources it would take a user a long time to find what they want just wading through a large list. Users need a way to search for resources quickly and find what they need by narrowing down their search results as much as possible.

### Project Objectives:

The objectives of this project are as follows…

To allow a user to capture 2 images from varying perspectives.

To allow users to narrow their search based on certain criteria such as by subject, authors name, date of publication, magazine/conference name etc.

To allow an authenticated administrator to securely log in and add, remove, or modify references while ensuring no duplicate exist in the database.

**Domain Analysis**

The domain of this project will be to develop a search engine accessing a database of contained resources which would include journal articles, publications, books, magazines and newspapers. The user interface of this document search engine will assist the user in finding documents in our database for purposes including but not limited to research, study, and personal interest.

The technology that will be needed to access our product will be computers. The project will provide services to many different types of people including but not limited to students, teachers, researchers, and anyone who is of an age old enough to understand the documents in the database.

# References

1. <http://www.visualstudio.com/en-us/products/visual-studio-ultimate-with-msdn-vs.aspx>
2. <https://gcc.gnu.org/>
3. <http://sourceforge.net/projects/argouml/>
4. <http://qt-project.org/>
5. <http://subversion.apache.org/>
6. <http://tortoisesvn.net/>

# Functional Requirements

Inputs

* Users should be able to search for resources by providing one or more keyword inputs. Inputs will be strings consisting of letters numbers and symbols (-.+,!,#,$).
* Searches can be based on a number of different criteria such as, subject, author name, title, type of document and publisher. (More than one attribute can be used to narrow the search).
* Users must authenticate with an individual log in to be able to access the system.
* Users will navigate the system through a simple and intuitive GUI (Graphical User Interface) to enable requests based on the above requirements.
* Users will be able to browse the database by alphabetical order, subject, authors name, date of publication, journal/conference name etc.
* System administrators will have access to administrative functions with their own individual authentication logins.
* Administrative functions will include the ability to “Insert Record” which adds a resource to the database, “Delete” which allows admin to delete a resource from the database, “Modify” which allows admin to modify the info of a resource, “Export to Text File” which allows the admin to export a copy of the database info to a text file on their local machine.

Outputs

* User information must be correct or an error is output to the user.
* Administrators will have the ability to export all resources and relevant data to an arbitrary file. Likely a .txt file.

Data

* The system will store resources in the database.
  + Resources will consist of books, magazine article and scholarly papers.
* User information will be encrypted and stored in the database.
  + User information includes email address, password, userID, username.
* Help files and tutorials will be stored in the database and accessible be users and administrators.
* The system must handle multiple clients (connections) concurrently and respond to each client request for services.

Use Case Diagram

|  |
| --- |
| **Brief Description**  The *Log in/Authenticate* use case enables users and admins to securely log on to the software. |
| **Step-By-Step Description**   1. User/Admin opens the program and is greeted with a clean U.I. login screen. 2. User/Admin enters their unique Username and Password into the appropriate text boxes. 3. User/Admin presses the login button and is brought to the main U.I. page. |

|  |
| --- |
| **Brief Description**  The *Search For Records* use case enables users and admins to search for resources located in the database. |
| **Step-By-Step Description**   1. User/Admin enters a their desired keywords into the search bar textbox. 2. User/Admin clicks on the search button. 3. User/Admin is taken to a page displaying the results of their search criteria and is able to scroll through the returned resources in a list. 4. User/Admin is able to narrow their search results by category in the sidebar. |

|  |
| --- |
| **Brief Description**  The *Insert Record* use case will enable the admin to add new resources to the programs database. |
| **Step-By-Step Description**   1. Admin clicks the “add new resource” button. 2. Admin is taken to the add new resource page which will have all the necessary fields represented with a textbox. 3. Admin will fill in the appropriate information in the textboxes. 4. Admin will click the “add resource” button and will be presented with a summary of their input and a button to accept or reject what they input. 5. Admin clicks accept and the resource is added to the software database. |

|  |
| --- |
| **Brief Description**  The *Delete Record*  use case will enable the admin to delete resources form the programs database. |
| **Step-By-Step Description**   1. Admin clicks the “remove resource” button and is taken to a new page with a search bar. 2. Admin searches for the resource they want to delete. (refer to search use class) 3. Admin clicks a checkbox to the left to the desired resource in the list. 4. Admin clicks the “remove resource” and is greeted with a summary of their action with a button to accept or reject |

|  |
| --- |
| **Brief Description**  The *Modify Record* use case enables admin to modify the information associated with a resource locate in the program database. |
| **Step-By-Step Description**   1. Admin clicks the “Modify Resource” button and is taken to a page with a search bar. 2. Admin searches for resource to be modified. (refer to search use class) 3. Admin is presented with a list of possible resources based on their search criteria. 4. Admin clicks the checkbox to the left of the resource that they desire t modify. 5. Admin clicks the “modify resource” button and is taken to a page with the current resource attributes in editable text boxes. 6. Admin enters any changes into the appropriate textbox. 7. Admin clicks the “accept changes” button and is shown a review of their changes and a button to either confirm or reject the changes. 8. Admin clicks “accept” and changes are made in the software database. |

|  |
| --- |
| **Brief Description**  The *Export To Text File*  use case will allow the admin to export a list of all resources in the database to a text file on their local machine. |
| **Step-By-Step Description**   1. Admin will click the “Export to Text File” button on the home screen. 2. Admin will be displayed a screen asking them where they want to export the file to on their local machine. 3. Admin is presented with a window asking them if they want to continue, they can select export or cancel. 4. Admin selects export and the text file is exported to their selected location. |

# Non-functional Requirements

Performance:

* The performance of the system will be determined at a later date based on variables such as database size, sorting/search algorithms, and UI constraints.
* Latency will be handled by displaying a “process has timed out” message after a determined maximum response time.
* Throughput:
  + This is the number of operations that can be performed per second.  It is still yet to be determined.
* Response Time:
  + The response time of the system has not been determined at this stage.

Quality Assurance:

* There shall be no duplicates of resources in the database.
* There shall be no duplicate user information.
* System administrator will receive an error whenever trying to enter duplicate information into the system.

Portability:

* The system will be limited to desktop PC and laptop.  Users will not be able to access the server with any mobile devices.

Robustness:

* Any errors will not be fatal to the operation of the program.
  + An error message will be displayed to the user/admin advising them of the error and what is needed to correct it.

Security:

* Password complexity will be enforced to protect against unwarranted access to the system.
* Document and user data will be encrypted as a second layer of security for the database. The type of encryption is to be determined.

Design Constraints:

* The number of resources displayed on a single page will be constrained to a certain number, to be determined later based on system performance.

Safety:

* The safety factor is not critical towards the user because it doesn’t deal with high voltages or complex control systems.

Privacy:

* User profile will be blocked and protected from any outside intruders trying to access the information.  The profile will include a user name and password, first and last name, home address, postal code and email address.

**Glossary**

*Administrator (Admin):* A person with higher level access to the software system, has more privileges. (refer to functional requirements)

*Database:* An organized collection of data that will be accessed by the user interface.

*Encryption:* the process of encoding messages or information in such a way that only authorized parties can read it.

*Latency*: a time interval between the stimulation and response from the software.

*Resource:* Any document contained withing the programs database

*Response Time:* the total amount of time it takes to respond to a request for service.

*Tools:* and piece of software used during the development of the project.

*User Interface (U.I.):* This includes everything user sees when they access the program, it is what the user will interface with. Also known as GUI (Graphical User Interface).

*Version Control:* An online repository that allows teams of people of organize code and keep a tractable history of changes. (ex. Git, Subversion, TFS)

**Case Tools**

The C.A.S.E. Tools that will be used in the development of this Software product will include…

* Visual Studio 2013 (Ultimate Edition)
  + V.S. will the main programming tool to be used for development of C++ code.
  + <http://www.visualstudio.com/en-us/products/visual-studio-ultimate-with-msdn-vs.aspx>
* GCC (GNU Compiler Collection)
  + GCC will be the compiler used to compile and run out C++ project code.
  + <https://gcc.gnu.org/>
* ArgoUML
  + ArgoUML is a UML design tool that will be used for creating diagrams including use case diagrams, UML diagrams, and sequence diagrams for this project.
  + <http://sourceforge.net/projects/argouml/>
* Qt
  + Qt is a cross-platform UI framework that we will be using for the development of our GUI.
  + <http://qt-project.org/>
* Apache Subversion
  + Subversion is the version control client that we will be using for this project, it will allow us to track changes and bugs as well as allow for a central location for all validated code.
  + It allows the development team members to have access to the latest project source code and to see any changes made to the code.
  + <http://subversion.apache.org/>
* TortoiseSVN
  + TortoiseSVN is the front-end SVN client that the team will use to keep out SVN up to date and track versions of the projects code.
  + <http://tortoisesvn.net/>

**(This page left intentionally blank)**