**Software Requirements Specification**

**for**

**Library Card Registration System**

**Version 1.0 approved**

**Prepared by Maurice Mejia**

**Alvin Tsui**

**Christopher Lee**

**Brent Hamada**

**Jose Aparicio**

**LA County Library**

**4/14/2018**

# Table of Contents

Table of Contents................................................................................................................... <pg 2>

Revision History..................................................................................................................... <pg 3>

1. Introduction................................................................................................................ <pg 4>
	1. Purpose........................................................................................................... <pg 4>
	2. Intended Audience and Reading Suggestions................................................ <pg 4>
	3. Product Scope................................................................................................ <pg 4>
	4. Definitions, Acronyms, and Abbreviations .................................................. <pg 5>
	5. References......................................................................................................<pg 5>
2. Overall Description.................................................................................................... <pg 6>
	1. Product Perspective........................................................................................ <pg 6>
	2. Product Functions...........................................................................................<pg 6>
	3. User Classes and Characteristics....................................................................<pg 7>
	4. Operating Environment.................................................................................. <pg 7>
	5. Design and Implementation Constraints........................................................ <pg 7>
	6. User Documentation...................................................................................... <pg 8>
	7. Assumptions and Dependencies.................................................................... <pg 8>
3. External Interface Requirements............................................................................... <pg 9>
	1. User Interfaces............................................................................................... <pg 9>
	2. Hardware Interfaces....................................................................................... <pg 9>
	3. Software Interfaces........................................................................................ <pg 9>
	4. Communications Interfaces........................................................................... <pg 10>
4. Requirements Specification....................................................................................... <pg 11>
	1. Functional Requirements............................................................................... <pg 11>
	2. External Interface Requirements................................................................... <pg 14>
	3. Logical Database Requirements.................................................................... <pg 15>
	4. Design Constraints......................................................................................... <pg 16>
5. Other Nonfunctional Requirements........................................................................... <pg 17>
	1. Performance Requirements............................................................................ <pg 17>
	2. Safety Requirements...................................................................................... <pg 17>
	3. Security Requirements................................................................................... <pg 17>
	4. Software Quality Attributes........................................................................... <pg 17>
	5. Business Rules............................................................................................... <pg 17>

Appendix A: Glossary............................................................................................................ <pg 18>

Appendix B: Analysis Models............................................................................................... <pg 19>

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  Init | 10/01/17 |  Initial changes and plan | 0.1  |
| First Draft | 10/26/17 | Complete Sections  | 0.2 |
|   | 10/27/17  | Revisions  | 0.2.1 |
| Second Draft | 11/7/17 | Completed Section 5 and other minor changes | 0.2.2 |
| Final Draft | 12/1/17 |  | 1.0 |

# 1. Introduction

This is the general overview for the LCRS

## 1.1 Purpose

This document will cover all the software requirements for the Library Card Registration System, otherwise known as the LCRS from here on out. This document will cover all the main modules and what they should all do. All the modules combined will make up the entirety of LCRS and how they interface with each other shall be detailed later. By the end of the document you should understand the main interfaces. You should have a general understanding of how the modules interact and their role in the whole registration system.

## 1.2 Intended Audience and Reading Suggestions

This documentation is intended for developers and testers. Developers can look through the documentation to understand the way the data will flow from client to database. As well as the internal structure of the modules and how they communicate. Testers can look to this document to see where in the interface are bugs coming from and which modules might be at the root cause for errors. In order to get a quick good understanding of the modules and interfaces, we suggest reading this document in the following order:

1. 2.3 - Product Functions
2. 2.4 - Operating Environment
3. 2.5 - Design and Implementation Constraints
4. 3 - External Interface Requirements

By reading the document in that order, you will get an idea of the project in terms of its behaviors, the design, and why it was designed and implemented the way it was.

## 1.3 Product Scope

The software product, known as the Library Registration Card System is also known as the acronym LCRS. The general goal of this software product is to allow users to register for the County of Los Angeles Public Library and services it provides. The main function of the LCRS is to update a database every time a user signs up, and registers through the LRCS. This process should be as easy as possible and there are additions to the LCRS whose purpose is help the user in filling out the form.

There are two modules whose purpose is to assist, and make filling out the form easier. The OCR module is in charge of providing users a way to upload an image that will be used to autocomplete part of the form and relevant fields. The Voice Assist module will help users fill in the application, using voice to populate the fields. It shall guide the user through the form by asking questions and using the user’s replies to autofill the correct field.

All of these modules, and a few more for verification and validation shall be used to create a form that is easy to use and enjoyable.

.

## 1.4 Definitions, Acronyms, and Abbreviations

**LCRS** - Library Card Registration System

**OCR** - Optical Character Recognition

**reCAPTCHA** - a CAPTCHA-like system designed to establish that a computer user is human (normally in order to protect websites from bots) and, at the same time, assist in the digitization of books.

**CAPTCHA** - Completely Automated Public Turing test to tell Computers and Humans Apart

**EULA** - End User Licence Agreement

**API** - Application programming interface

**WCAG** - Web Content Accessibility Guidelines

**GUI** - Graphic User Interface

## 1.5 References

There are no references in this document

# 2. Overall Description

This section shows the general descriptions about the LCRS.

## 2.1 Product Perspective

This software will rely on a few other components or software services. For the text recognition we will be using Google Cloud Vision API. Cloud Vision is an image recognition library, maintained and developed by Google. It will be the main software employed to assist in image processing and recognition. The OCR module will be a service that will be used to process images. Users will be able to upload an image from the form to use the OCR module.

We will use various Google services, including the Google Maps API along with Google’s CAPTCHA services. These will be used within our modules, mainly for validation.

As the user completes the form whether through text entry, or through one of the other methods we will need to verify and validate that data. The two most important fields that require validation are the user email address, and street address.

Aside from the services from Google that will assist us with validation and provide us with a starting point for image processing this will be a standalone product.

## 2.2 Product Functions

This product will have one major function. The main function is described as allowing users to sign up for County of Los Angeles Library services. The modules that help support the main function are described below:

* GUI
	+ The Graphic User Interface (GUI) will allow a user to enter data, and submit into the County of Los Angeles Public Library endpoint. It will allow users to enter data through regular text input, or via an image of a driver's license, or voice. This form shall also validate data entered by the user. It shall use client side validation and server side validation to make sure that there are as few errors in the data as possible.
* Voice Assist
	+ The Voice Assist is to assist users with disabilities to fill out the application. It will provide voice feedback for each component of the application. It will also provide voice input so that the user will be able to fill out the application.
	+ This module serves two purposes: to guide users through the form and to make it easier for users with disabilities to fill out the form by reading it out, making the entering of data easier.
* OCR
	+ The Optical Character Recognition (OCR) is responsible for giving the option to scan their driver’s license to fill in their home address for them. This module enables image upload from hard drive on computer or from phone camera. Once an image is received OCR libraries and software shall extract relevant data and autofill into the appropriate fields. If the OCR cannot ensure accurate data extracted from the image, the data shall be rejected and the field shall remain empty.
* Address Lookup
	+ This module is in charge of validating the user’s entered physical street address. This module shall connect with the appropriate Google services to autofill and autocomplete the addresses.

These are the main functions of LCRS. The overall goal, and function of LCRS is to submit data entered by the user to the County of Los Angeles Public Library endpoint and create a simple but enjoyable experience when registering for library services.

## 2.3 User Classes and Characteristics

The LCRS will be used by various types of users. The overall user should be able to use the LCRS without any difficulty. Our target user is someone who may not be technologically savvy, and may speak a different language other than English. We must make sure to keep these users in mind in the design and implementation of LCRS

## 2.4 Operating Environment

The LCRS will be built using a variety of tools. The main application will be a Django-based web application. The application web interface will be implemented by Django web framework. This will be connected to other Django web applications that will be used to assist in entry and validation, as well as submission. This project shall be hosted on a Linux server, within a contained Python environment (Virtualenv) to help eliminate errors caused by dependency and software version issues. The LCRS will work on any platform as long as the user is using a supported browser.

## 2.5 Design and Implementation Constraints

The LCRS must be accessible for users using various devices. This means that it must be implemented using responsive design. It must also be accessible by other accessibility requirements, such as Web Content Accessibility Guidelines (WCAG). The LCRS should be implemented with the appropriate HTML elements to provide a clear description of the content, and it must also be accessible by various languages. The LCRS will be able to support a wide range of browsers.

## 2.6 User Documentation

WCAG 2.0: <https://www.w3.org/TR/WCAG20/>

## 2.7 Assumptions and Dependencies

The LCRS will have a few dependencies:

* Google Maps API
	+ <https://developers.google.com/maps/>
* Google reCAPTCHA
	+ https://developers.google.com/recaptcha/intro
* Google Cloud Vision API
	+ <https://cloud.google.com/vision/>
* USPS Web Tools API
	+ <https://www.usps.com/business/web-tools-apis/welcome.htm>

#

# 3. External Interface Requirements

This section will describe how the LCRS will interact with external interfaces.

## 3.1 User Interfaces

The user interface, or GUI, will consist of a view with may input fields. It will implement responsive design to display the form for various screen sizes. There will be a button that will turn on the Voice Assist feature and another button that will submit the entry into the Main module. It will be a single page, and it should be easy to read.

## 3.2 Hardware Interfaces

The Voice assist module will require the use of a microphone and speakers. The microphone will be used to capture voice for voice to text entry. Speakers will be needed so the user can hear the voice assist module’s requests. This hardware can be found on most computers and phones. Browser support for access to the hardware will vary as different browsers will use different implementations. The OCR module requires camera access, again access to this hardware will vary with browser. LCRS will use the appropriate browser level API to gain access to required hardware.

## 3.3 Software Interfaces

The LCRS will require access to the Library’s own private database in order to create Library Card users using the required fields in the application that were filled in by the users. The information requested will be identical to that required by the library’s paper registration form. In its testing version, the LCRS will be storing the information into its own database. However, once it gets finalized, the LCRS will only be making API calls to read and write to the Library’s own database. When LCRS is deployed in production environment, it shall use County of Los Angeles Library database and their API. However, during development and for the staging environment we will have a staging database to test the application.

## 3.4 Communications Interfaces

Once the user interacts with the LCRS and makes a submission, the LCRS will then send an email to the entered email address in the application to ensure that the said email address is valid. This email will, at minimum, contain a button that will direct the user to an HTML webpage on the Library’s website that will confirm the validity of the entered email address in the LCRS once the button is clicked on.

#

# 4. Requirements Specification

This section contains the software requirements for the application. The requirements will be detailed enough so that designers and developers will be able to accurately fulfill these requirements. The technical requirements, include how the modules and requirements will be fulfilled during development.

## 4.1 Functional Requirements

|  |
| --- |
| Requirements Related to Design Module 4.1 GUI |
| Requirement No. |  |
| 4.1-1 | The GUI shall consist of input elements for: First name, Middle name, Last name, Address 1, Address 2, City, State, Zip code, Apartment number, Residence address, Birthday, Phone number, California Driver License Number, last four digits of Social Security Number, E-mail, Electronic signature. |
| 4.1-2 | The GUI shall have a Voice Assist function that will read all text field prompts |
| 4.1-3 | The GUI shall use the Google Translate API to translate application language to languages that are not English or Spanish |
| 4.1-4 | The GUI shall have an Email Confirmation function |
| 4.1-5 | The GUI shall have a submit button that will send the user input to a data base. |
| 4.1-6 | The GUI shall have image upload option that will allow users to upload an image of their drivers license |
| 4.1-7 | The GUI shall have an Address validation that will confirm if the Address is located in California |
| 4.1.8 | The GUI module shall ensure that the application will comply with the WCAG requirements. |
| 4.1-9 | The GUI shall have a reCAPTCHA checkbox |

|  |
| --- |
| Requirements Related to Design Module 4.1.2 Voice Assist |
| Requirement No. |  |
| 4.1.2-1 | The Voice Assist shall be able to guide user through input fields |
| 4.1.2-2. | The Voice Assist shall automatically fill fields based on user voice response |
| 4.1.2-3 | The Voice Assist shall confirm each input field with user |

|  |
| --- |
| Requirements Related to Design Module 4.1.3 Optical Character Recognition (OCR) |
| Requirement No. |  |
| 4.1.3-1. | The OCR shall enable access to the camera of the user’s phone or allow an image type file upload.  |
| 4.1.3-2. | The OCR shall scan an uploaded image file of a driver’s license to recognize the characters. |
| 4.1.3-3 | The OCR shall be able to fill for the following fields: First Name, Last Name, Middle Initial, Birthdate, and Address. |
| 4.1.3-4 | The OCR shall determine which data is associated with which field. An address indicated on a driver’s license shall be directed to the address field. |
| 4.1.3-5 | The OCR shall communicate with the Address Lookup module to verify the address field filled by OCR. |

|  |
| --- |
| Requirements Related to Design Module 4.1.4 Email Confirmation |
| Requirement No. |  |
| 4.1.4-1 | The Email Confirmation shall send an email to user after submission to verify email |
| 4.1.4-2 | The Email Confirmation shall send unique codes to each email provided |
| 4.1.4-3 | The Email Confirmation shall store these unique codes in the database for each user. |
| 4.1.4-4 | The Email Confirmation shall update the database if the user does verify the user’s email by comparing the code the user provides, with the code in the database. |
| 4.1.4-5 | The Email Confirmation shall have an API that the link will make a request to that will update, and tag user as valid. |
| 4.1.4.6 | The Email Confirmation module shall make calls to the Library API to record the data provided |
| 4.1.4.7 | The Email Confirmation module shall generate a login number after the user clicks the confirmation URL link in their email |

|  |
| --- |
| Requirements Related to Design Module 4.1.5 Address Lookup |
| Requirement No. |  |
| 4.1.7-1 | The Address Lookup shall query Google Maps API with user input or OCR input. |
| 4.1.7-2 | The Address Lookup shall accept only valid California street addresses. |
| 4.1.7-3 | The Address Lookup shall return an error if a valid California street address is not found. |
| 4.1.7-4 | The Address Lookup shall automatically complete form fields with the validated information obtained from Google Maps API |

## 4.2 External Interface Requirements

This section will go through the many modules in this application and describe in detail the inputs and outputs of each. By the end of this section you should have enough of an understanding of the application that you would be able to design something based on these descriptions.

**Main Module**

* The main module will be the main application program that interfaces with all of the other modules of the application. It consists of a Python program that will make calls to the appropriate modules.
* The main module will handle both input and output. It will call the correct modules with the appropriate arguments. Once the called module is finished, the main module will receive a response upon success. The source of input for the Main module will be user input such as clicks, and text entry.

**GUI**

* The GUI module is the main view for the user. This is the entry point to the application. It consists of a website that will serve as a view. It also has multiple options for selecting how the user will input data to the application.
* Data input will come from the user from GUI actions such as text entry, mouse clicks, or voice commands. The text entry functionality will allow users multiple ways of completing the application. This module can send data to any modules at any time during user input. Once data has been retrieved from the other modules and the user is ready to submit the data, the browser shall submit a post request and the application will update the database.

**Voice Assist**

* This module will be one of the main ways that a user can input text into the application. This module will guide users through the application and use voice input to complete the application. It will be powered by a speech-to-text API. Our module will leverage this API by using it to convert user response to text. This text will then be used to auto-fill the form. We will also be using a text-to-voice library. A function will use this library to ask the user questions so that the text-to-voice library can convert the response.
* In order to use this module, the user will need to invoke the Voice Assist module. This will happen through the user interface that will display a button to invoke this feature. After that there will be a series of outputs in the form of text-to-voice questions, and inputs in the form of user replies.

**OCR**

* This module will take in an image and using computer vision libraries will extract text from the image. This text will then be used to fill the relevant fields. In order to get accurate results from the image there will be certain requirements that the user needs to meet. Image requirements will be detailed in the documentation. The image will be supplied via a browser API. The image will not be saved on our servers. The data will be processed and then sent to the application. Neither the image, or data will persist in the application once submission has happened.
* An image will be taken as input and the output will be text. The output will go to the application, or GUI module.

**Email Confirmation**

* This module will send a user a link by email. Once that link is clicked we will flag the user as a valid user. This module will be called open submission of the application. When the user clicks the link, we will have an API that will respond that to request by updating our database and setting the appropriate flag.

## 4.3 Logical Database Requirements

The LCRS will only store data to a test MySQL database. It will currently store these attributes:

* id (default)
* Last Name
* First Name
* Middle Name
* Mailing Address
* Apartment Number
* City
* State
* Zip Code
* Residence Address
* Telephone
* Birthdate
* California Driver’s License
* Last Four Digits of Social Security Number
* Email
* How to notify the user (by mail, email, or telephone)
* First Name of Parent or Guardian (for minors)
* Last Name of Parent or Guardian (for minors)
* Address of Parent or Guardian (for minors)
* Date (for minors)
* Movie Access (for minors)
* Email Code

## 4.4 Design Constraints

The application will be hosted on a standard Linux server. The client side shall function correctly on the following browsers using the WCAG 2.0 standards:

* Chrome 60+
* IE 11+/Edge14+
* Firefox 54+
* Safari 9+
* Opera 46+

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

The application will support these performance requirements:

* The rendering of the client side HTML shall be completed in no more than 2 seconds.
* It shall support multiple users at the same time.
* Processing of the data on the server shall be completed in no more than 2 seconds.

## 5.2 Safety Requirements

LCRS does not have any safety requirements.

## 5.3 Security Requirements

LCRS needs to handle various cases such that it is a robust application. It needs to handle DDOS such that the application cannot be brought offline from a malicious attack. Form fields need to have server side validation such that SQL injection attacks and such won’t be effective against LCRS. LCRS will require a SSL certificate for encrypting http requests and responses.

## 5.4 Software Quality Attributes

LCRS needs to be useable by a variety of devices including but not limited to smartphones, tablets, and varying sized monitors. It needs to implement responsive design to make sure the application is usable and appears elegant on all supported devices. LCRS also needs to be easy to use. That means a user should experience no difficulty completing the form. The tags and labels of the form should be self-explanatory for the supported languages.

## 5.5 Business Rules

All users who register with LCRS shall be required to use a valid California address. Users will also be required to disclose if they are of age, or if parent consent is required.

#

# Appendix A: Glossary

LCRS - Library Card Registration System

OCR - Optical Character Recognition

Voice Assist - Voice Assist

reCAPTCHA - a CAPTCHA-like system designed to establish that a computer user is human (normally in order to protect websites from bots) and, at the same time, assist in the digitization of books.

CAPTCHA - Completely Automated Public Turing test to tell Computers and Humans Apart

EULA - End User License Agreement

API - Application programming interface

WCAG - Web Content Accessibility Guidelines

# Appendix B: Analysis Models



Figure 1. Level 0 DFD



Figure 2. DFD Level 1