Software Requirements Specification

UMGC CITY APPLICATION

**Version 1.0**

**Prepared by UMGC City Team 1**

**Christy Gilliland**

**Daniel Abresch**

**Jack Amnuaysirikul**

**Krystina Poling**

**Melanie Meek**

**Patience Okereke**

**Tarig Abasit**

**Ziad Elharaoui**

**03/20/2020**

Contents

[1 Introduction 4](#_Toc35439568)

[1.1 Purpose 4](#_Toc35439569)

[1.2 Intended Audience 4](#_Toc35439570)

[1.3 Product Scope 5](#_Toc35439571)

[1.4 Definitions, Acronyms, and Abbreviations 6](#_Toc35439572)

[1.5 References 7](#_Toc35439573)

[1.6 Overview 7](#_Toc35439574)

[2 Overall Description 7](#_Toc35439575)

[2.1 Product Perspective 7](#_Toc35439577)

[2.2 Product Functions 9](#_Toc35439578)

[2.3 User Classes and Characteristics 10](#_Toc35439579)

[2.4 Constraints 10](#_Toc35439580)

[2.5 Assumptions and Dependencies 10](#_Toc35439581)

[2.6 Apportioning of Requirements 11](#_Toc35439582)

[3 Specific Requirements 13](#_Toc35439583)

[3.1 External Interface Requirements 13](#_Toc35439585)

[3.2 System Features/Modules 17](#_Toc35439586)

[3.3 Design and Implementation Constraints 33](#_Toc35439589)

[3.4 User Documentation 34](#_Toc35439590)

[4 Nonfunctional Requirements 35](#_Toc35439591)

[4.1 Performance 35](#_Toc35439592)

[4.2 Security 35](#_Toc35439593)

[4.3 Quality 35](#_Toc35439594)

[4.4 Cultural, Political and Accessibility 36](#_Toc35439595)

Table of Tables

[Table 1 - UMGC City Team 1 Roles 4](#_Toc35439596)

[Table 2 - Definitions, Acronyms and Abbreviations 6](#_Toc35439597)

[Table 3 - Software Interface Description 15](#_Toc35439598)

[Table 4 - Agile User Stories 38](#_Toc35439599)

### Version History

|  |  |  |
| --- | --- | --- |
| **Date** | **Reason For Changes** | **Version** |
| 03/12/20 | Initial Version  Approved:  Christy,  Reference your email below dated 3/12/2020 5:48:45 PM Eastern Standard Time, Subject: UMGC City Team 1 SRS.  Reviewed and approved.  Thank you,  Roy Gordon  Looks great, reviewed and approved. APIs work out?  Thanks,  Israel | 1.0 |
| 03/18/20 | Edited version per Dr. Mir’s Request and Guidance | 2.0 |

# 1 Introduction

The following subsections provide a general overview of this document, formally known as the Software Requirements Specification (SRS). These subsections also present a top-level description of the UMGC City Application. The information here was excerpted from the project plan approved by the stakeholders on February 22, 2020.

## 1.1 Purpose

This document describes the functional and nonfunctional software requirements for version 1.0 of the UMGC City Application. This document acts as a contract of system requirements and is intended to be used by the members of UMGC City Team 1 to implement the stated requirements. This document will also be used by the stakeholders to verify that the application has been built to specifications. Unless otherwise noted, all requirements specified here are high priority and committed for version 1.0.

## 1.2 Intended Audience

The intended audience of this document includes all the members of UMGC City Team 1 and the project stakeholders. Table 1 defines the roles within the team.

Table 1 - UMGC City Team 1 Roles

| **Name** | **Primary Role** | **Secondary Role(s)** |
| --- | --- | --- |
| Israel Del Toro | Client | Stakeholder |
| Ray Gordon | Project Manager | Stakeholder |
| Dr. Mir Assadullah | Professor | Stakeholder |
| Christy Gilliland | Project Manager | Developer, DBA |
| Tarig Abasit | Developer | Requirements Analyst, DBA |
| Daniel Abresch | Developer | Tester, DBA |
| Jack Amnuaysirikul | Tester | Requirements Analyst, DBA |
| Ziad Elharaoui | Lead Developer | Requirements Analyst |
| Melanie Meek | Requirements Analyst | Tester |
| Patience Okereke | Requirements Analyst | Tester |
| Krystina Poling | Lead Developer | Tester/DBA |

## 1.3 Product Scope

The scope of the UMGC City project is to build a free, open-source application that can be used by city officials to improve the usability of existing city web portals. The objective of the application is to enhance the user experience by helping them easily and intuitively locate applicable city ordinances for a predetermined list of frequently requested user inquiries (as determined by the city officials). In order to accomplish this goal, the application shall present the city officials with an interface to accept input of various use cases along with all the pertinent information that will benefit their intended audience. The application shall process that data into appropriate tables that comprise a database. When the users visit the city web portal to search for specific ordinances, the database shall be called upon to produce appropriate output that is tailored to them in a user-friendly format. This generic, build-to-suit database infrastructure allows the application to be reused by any city.

The UMGC City teams, along with the DevOps team, will work closely with the customer, Israel Del Toro, to design the database. In addition, the development teams shall tailor the application to deliver new pages and functionalities that are customized for the website of the City of Pasadena, California. Specifically, the UMGC City project teams will be delivering two interfaces: an interactive map/web-based interface (Team 1) and a ChatBot interface (Team 2). This SRS document presents all system requirements applicable to the database setup and the map/web-based interface, whose tasks fall under UMGC City Team 1. UMGC City Team 2 will use the same database designed by Team 1 to ensure the same data is used for both interfaces.

## 1.4 Definitions, Acronyms, and Abbreviations

Table 2 - Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **ChatBot** | a computer program designed to simulate conversation with human users |
| **CRUD** | “Create, Read, Update and Delete” |
| **DevOps** | Development and Operations Team |
| **DFD** | Data flow Diagram |
| **ERD** | Entity Relationship Diagram |
| **GUI** | Graphical User Interface |
| **HTTP Request** | The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers. HTTP works as a request-response protocol between a client and server. |
| **PostgreSQL** | A general-purpose and object-relational database management system. |
| **REST Controller** | This is used to create RESTful web services using Spring MVC. It takes care of mapping request data to the defined request handler method |
| **REST API** | an application program interface ([API](https://searchapparchitecture.techtarget.com/definition/application-program-interface-API)) that uses HTTP requests to get, put, post and delete data |
| **Spring Data JPA** | Part of the larger Spring Data family. This makes it easy to implement JPA based repositories |
| **SRS** | Software Requirements Specification |
| **UMGC** | University of Maryland Global Campus |
| **PaaS** | Platform as a Service - a cloud computing model in which a third-party provider supplies the software and hardware infrastructure required to develop, run, and manage an application. |

## 1.5 References

* https://learn.umuc.edu/d2l/le/content/444089/viewContent/16973490/View (Assignment Description)
* IEEE Recommended Practice for Software Requirements Specifications. IEEE Std 1058.1-1987, 31 Aug. 1988. Accessed via <http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf>

## 1.6 Overview

The SRS will provide a comprehensive description of the intended purpose and environment for the UMGC City Application. The final approved version of this SRS will be used at delivery to measure the success and viability of the project.

Section 2 of this document provides an overall description of the UMGC City Application, the associated interfaces and user classes. Section 3 provides specific development requirements, including operating environment, database construction and user support. Section 4 contains a detailed description of each system feature, including functional requirements for each, and a list of nonfunctional requirements for the entire application.

# 2 Overall Description

This section provides an overall description of the UMGC City Application, the associated interfaces and user classes. It is a top-level representation of the expected final deliverable.



## 2.1 Product Perspective

The UMGC City Application is a collection of interfaces designed to assist city officials in creating a database and a customer-facing HTML interface. The customer-facing HTML interface is created from a pre-configured HTML template and data collected from city users and stored in the application’s database. The result of the application deliverable is a static HTML page in an easy-to-navigate format for use by city residents.

The UMGC City Application is intended to integrate with current websites employed by the cities. This SRS and the development of the UMGC City Application does not imply suitability to wholly replace current systems.

### 2.1.1 User Interfaces

The UMGC City Application’s user interface shall follow four basic design principles to ensure the system’s UI is intuitive, efficient, and easy-to-navigate for a wide range of users. The four design principles that will be followed are simplicity, structure, consistency, and tolerance. This means the UI design will be kept modern but minimal to ensure the UI does not become confusing to users or over-crowd the user’s screen with unnecessary content. The UI will follow a basic and logical data gathering process using structured data input forms. In addition, the entire design of the UI meaning color schema, behavior, and content placement shall be kept consistent throughout the entire web application. Lastly, the UI will aid the user in preventing mistakes by confirming all input decisions before acceptance into the system is permitted and provide backward error recovery whenever applicable.



### 2.1.2 Hardware Interfaces

The UMGC City Application requires an up-to-date computer system with a modern browser and an active internet connection. The application shall be deployed and maintained on the cloud application platform Heroku, also known as a Platform-as-a-Service (PaaS) cloud computing architecture and shall have no specific hardware interface requirements.

### Software Interfaces

The UMGC City Application is expected to produce a static HTML page available to citizens that can be integrated with a city’s current web application architecture. In addition, the static HTML deliverable will link to third-party web pages that shall be specified by the city user at time of creation. Additional integration with external software interfaces is also required for the UMGC City Application to be built, deployed, and maintained.

## 2.2 Product Functions

The following list shows the product functions as applicable to city management:

1. Create an account (username and password registration are required).
2. Create and define a use case.
3. Save to database.
4. Load use cases to the database using a comma-delimited file (CSV).
5. Manage/edit use cases and zone information in the database.
6. Delete use cases and zone information in the database.
7. Publish the Quick Reference Use Case Page to an HTML file.

The following list shows the product functions as applicable to city residents:

1. Explore custom map of city zones and districts.
2. Click on a specific zone on the map to display tailored information for each zone.
3. Navigate to the Quick Reference Use Case page.
4. Explore more information regarding specific use cases as presented by the city management.
5. Links navigate to an external source, e.g., <https://library.municode.com/ca/pasadena/codes/code_of_ordinances>.

## 2.3 User Classes and Characteristics

The system will support two types of user privileges, city resident and city official.

|  |  |
| --- | --- |
| User-1: | The user is a city resident who intentionally navigates to the city’s web page in search of zoning information and applications. |
| User-2: | The user is a city official who maintains zoning information and applications for a city. |

## 2.4 Constraints

The application is web-based and thus requires the user to have an updated browser and active internet connection. Chrome is the recommended and targeted browser for this application.

## 2.5 Assumptions and Dependencies

* It is assumed that the cities utilizing this application will appoint an official who is familiar with web-based applications. The official will be educated in their respective city ordinances and be able to provide correct information for each desired use case.
* It is assumed that the end users (city residents) are familiar with how to use web-based applications and are familiar with the use of a map.
* It is assumed that the user has a computer system which can handle the resources the application requires.
* It is assumed that the user has a stable internet connection to access the application.
* It is assumed that the client has the resources to host the application in an Apache Tomcat web server and a PostgreSQL RDBMS.

## 2.6 Apportioning of Requirements

Version 1.0 of the UMGC City Application is available only in English. Customized city maps are not included in the application. Future versions should implement these features to provide a greater level of usability for other cities.

The Version 1.0 deliverables for the City of Pasadena, California include a customized map and a pre-formatted Quick Reference page. The use cases identified by the client to be included in the database on delivery are as follows:

1. **Allowed Land Uses (Parcel Specific):**
   1. Select zone via map interface
   2. Retrieve all land uses associated with their parcel specific zone code
   3. Retrieve definition of land use
   4. Access entitlement process and requirements (e.g. Conditional Use Permit) if required for land use
   5. Access link to application
2. **Home Occupation Permit:**
   1. Retrieve regulations
   2. Access links to application
3. **Accessory Dwelling Unit:**
   1. Retrieve development standards
   2. Access application via link
4. **Short Term Rental:**
   1. Retrieve regulations
   2. Access Host Compliance application via link
5. **Accessory Structure:**
   1. Retrieves regulations
   2. Access zoning application via link
6. **Exterior Modifications in Landmark District:**
   1. Enter address
   2. Retrieve information that identifies whether a property is in a landmark district
   3. Retrieve regulations as to what projects are subject to a Certificate of Appropriateness (CofA)
   4. Access CofA application via link
7. **Fence:**
   1. Retrieve explanation of what constitutes front yard/side yard/backyard for non-traditional lots
   2. Retrieve development standards for lots
8. **Tree Removal (Residential)**
   1. Retrieve information defining “what is a protected tree?”
   2. Retrieve criteria for determination: specimen, diameter at breast height (DBH) and location
   3. Retrieve eligibility criteria for removal
   4. Access link to PLN application for removal
9. **Day-Care**
   1. Retrieve definition of small, medium and large day cares
   2. Retrieve development standards for each category
   3. Access link to entitlement process where applicable (e.g. CUP for large day cares in residential zones)
10. **Sober Living Facility**
    1. Retrieve definition of sober living facility
    2. Retrieve regulations with operation
    3. Access link to Reasonable Accommodation Permit Application

# 3 Specific Requirements

This section provides specific development requirements for the operating environment, database construction and user support.



## 3.1 External Interface Requirements

The following subsections explore the requirements of the external interfaces of the UMGC City Application.

### User Interfaces

The system’s UI navigation requirements to move from one UI screen or component to the next are listed below.

UI-1.1 The UMGC City Admin Portal UI shall display a Home Page Screen.

UI-1.2 The UMGC City Admin Portal UI shall display a Sign-in Screen.

UI-1.3 The UMGC City Admin Portal UI shall display a Sign-Up for new users.

UI-1.4 The UMGC City Admin Portal UI shall display a Help Screen.

UI-1.5 The UMGC City Admin Portal UI shall display a Landing Page Screen to the user when the user inputs valid login credentials into Sign-in Screen.

UI-1.6 The UMGC City Admin Portal UI shall display the New Use Case Screen to the user when the user selects the “New Use Case” option.

UI-1.7 The UMGC City Admin Portal UI shall display the File Upload Screen to the user when the user selects the “File Upload” option.

UI-1.8 The UMGC City Admin Portal UI shall display the Manage Existing Use Case Screen to the user when the user selects the “Manage Existing Use Case” option.

UI-1.9 The UMGC City Admin Portal UI shall display the Delete Use Case Screen to the user when the user selects a use case and selects the “Delete Use Case” option.

UI-1.10 The UMGC City Admin Portal UI shall display the Export Project Screen to the user when the “Export Project” option is selected on the Manage Existing Use Case Screen.

UI-1.11 The UMGC City Admin Portal UI shall display the Generate Output Screen to the user when the “Generate output” option is selected on the Export Project Screen.

UI-1.12 The UMGC City Admin Portal UI shall display the Display Preview Screen to the user when the “Display Preview” option is selected on the Export Project Screen.

UI-1.13 The UMGC City Admin Portal UI shall display the Download Project Screen to the user when the “Download” option is selected on the Export Project Screen.

### 3.1.2 Software Interface

There are multiple software components that the application must interact with to perform the stated requirements. External software requirements are listed in Table 3.

Table 3 - Software Interface Description

| **Software** | **Description** |
| --- | --- |
| Operating system | · Windows 7, Windows 8, Windows 8.1, Windows 10 or later  · Servers require Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, or Windows Server 2016  · OS X Yosemite 10.10 or later  · 64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+ |
| Database | PostgreSQL (Free/Open source)  Version 12 and later |
| Java | Java 11, all releases |
| Tomcat Web Server | Version 9 or later  Source:<https://tomcat.apache.org/whichversion.html> |
| Apache Maven | Version 3.6.3, all releases |
| Angular | Version 9, all releases  Angular CLI 9.0.0-rc.3  Typescript 3.7 |
| npm/node.js | Node 10.13 or later  npm latest version |
| IntelliJ | Version 2019.3.3 or later |
| Visual Studio Code | Version 1.31 or later |
| Google Chrome | Version 80 or later  Source:<https://www.google.com/chrome/?brand=CHBF&ds_kid=43700012290718061&gclid=CN6h39iEk-gCFVIAiAkdr9ULtA&gclsrc=ds> |
| HTML | HTML5, map.css and map.js |
| Misc. Supported Browsers | Firefox, Edge, Safari 9+ |
| Python | Python 3.7.7 |
| PyCharm | Version 2019.3.1 or later |
| Spring Boot | Version 2.1 |

Requirements for software interface integration are defined below:

SI-1.1 The system shall interact with PostgreSQL RDBMS for all data transactions.

SI-2.1 The system's server-side Java EE application shall interact with the system's client-side Angular web application.

SI-3.1 The system’s client-side Angular application shall interact with the system’s server-side Java EE application.

SI-4.1 The system’s client-side Angular application shall interact with the file storage system of the user’s device.

SI-5.1 The system’s server-side Java EE application shall interact with a Python script to parse CSV data files.

SI-6.1 The system shall interact with a Chatbot Java web application developed by UMGC City Team 2.

SI-7.1 The system shall be able to build, deploy and maintain a relational database to store city information.

### 3.1.3 Operating Environment

OE-1.1 For increased accessibility, the application shall operate on a web server compatible with a common, modern browser.

OE-2.1 The application shall authenticate users using a secure validation process and Hypertext Transfer Protocol Secure (HTTPS) requests.

OE-3.1 The system shall be able to build, deploy and maintain a relational database to store city information.

## 3.2 System Features/Modules

The following subsections provide an in-depth view of the system features. Each subsection focuses on a specific feature and presents a series of applicable stimulus/response pairs and functional requirements.

### 3.2.1 Database Application for City Officials

This is a database application with a generic system design approach that is free and open to use for all city officials.

### 3.2.1.1 Description and Priority

The goal is to create a web page with tailored information for the most frequent land use inquiries. The application shall work as a city admin portal, serving as a one-stop shop for city officials to help them create custom project use cases in order to improve the usability of their existing city websites. The use cases can be tailored to a wide range of audiences and thus shall be effective in enhancing the overall user experience. The resulting web page shall have all applicable city definitions, regulations, ordinances, permit requirements, and applications. Priority = High.



### 3.2.1.2 Stimulus/Response Sequences

Stimulus: The user enters the URL of https://www.umgccity.com into a web browser.

Response: The user is presented with the home page of the web application, UMGC City Admin Portal.

Stimulus: A user clicks on the “Sign-In” button.

Response: The system brings the user to the sign-in page, asking the user to input email address and password to sign in before granting access to all features of the application.

Stimulus: The user clicks on the “Sign-Up” button.

Response: The system brings the user to the sign-up page, prompting the user to enter data into all the required fields before submission to create a new city admin account.

Stimulus: The user clicks on the “Help” button.

Response: The system brings the user to the help page that displays all the help options which are available to the user.

Stimulus: The system successfully authenticates the user sign-in.

Response: The user is presented with the landing page of UMGC City Admin Portal. Once previously locked features are now available to the user. These include, but not limited to, the ability to create a new use case, edit existing use cases, and upload a use case.

Stimulus: The user clicks on the “New Use Case” button.

Response: The system presents the user with the ability to create a new use case. The user is asked to fill in all the necessary information with an option to save the use case in the database.

Stimulus: The user clicks on the “Existing Use Case” button.

Response: The system presents the user with a list of all use cases associated with his/her account. Here the user can select a use case from the list to continue working on that use case.

Stimulus: The user clicks on the “Upload Use Case” button.

Response: The user is brought to a page where he/she can upload use cases using external files that follow specific formats and standards.

Stimulus: The user clicks on the “Edit” button.

Response: The system brings the user to a page where he/she can edit the previously saved data of the selected use case.

Stimulus: The user clicks on the “Delete” button.

Response: The system prompts the user to confirm the deletion of the selected use case. Once confirmed, that use case is deleted, and the list of all use cases is updated to reflect the change.

Stimulus: The user clicks on the “Browse” button.

Response: The system opens a separate Windows frame, prompting the user to locate an external file within the computer directory to be uploaded.

Stimulus: The user clicks on the “Upload” button.

Response: The system shall process the user request to upload the selected file. Here the use cases will be automatically populated and added to the use account if the file passes the requirement formats and standards.

Stimulus: The user clicks on the “Project Export” button.

Response: The system brings the user to the project export page where directions are given to the user to perform specific functions associated with exporting the project.

Stimulus: The user clicks on the “Generate Output” button.

Response: The system processes the project data and generates output consisting of static HTML page(s) and appropriate DDL script(s) that is ready to be exported.

Stimulus: The user clicks on the “Preview” button after the output has been generated.

Response: The system opens a separate Windows frame that contains a display preview of the application output. Here the user can ensure the correct look and feel by seeing how their static HTML page(s) is displayed.

Stimulus: The user clicks on the “Download” button after the output has been generated.

Response: The system opens a separate Windows frame, prompting the user to choose where within the computer directory to save the output file for project export.

Stimulus: The user clicks on the “Submit” button.

Response: The system accepts data input by the user for processing.

Stimulus: The user clicks on the “Back” button.

Response: The user is brought back to a page that he/she is previously on.

Stimulus: The user clicks on the “Save” button.

Response: The system processes data inputted by the user. If the entries are valid, the user data is saved onto the account.

Stimulus: The user clicks on the “Sign-Out” button.

Response: The system safely logs the user out of the application.

Stimulus: The user clicks on the “Home Link” logo.

Response: The user is either brought to the landing page if logged in or the home page if currently not signed in.

Stimulus: The user clicks on specific hyperlinks

Response: The system shall bring the user to the appropriate page based on the hyperlink context. For example, if the user clicks on “First Time Here? Click Here to Create an Account,” the system brings the user to the sign-up page.

### 3.2.1.3 Functional Requirements

REQ-1.1: The home page of UMGC City Admin Portal system shall be accessible to the user via the following web URL: https://www.umgccity.com. On this page, the user shall be presented with necessary information (app description, purpose, instructions, etc.) to help him/her get started.

REQ-1.2: There shall be a consistent main navigation bar along the top of web pages to allow the user to quickly navigate within the application. Some of the navigation options within the bar are only accessible when the user is logged into the account because the use of the application is intended for city officials. Also, there are hyperlinks embedded within the application to help the user reach proper destinations, e.g., “First Time Here? Click Here to Create an Account.”

REQ-1.3: The UMGC City Admin Portal system shall provide a new user with the ability to sign up for an account by clicking on the “Sign-Up” button. The user is prompted to enter appropriate information to indicate that he/she is a city official before an account can be created.

REQ-1.4: The UMGC City Admin Portal system shall allow a user to sign into his/her existing account by clicking on the “Sign-In” button. Once the user submits the email address and password, the system shall authenticate the log-in. The user shall be given the full features of the application when signed in successfully. In addition, the ‘Sign-In’ button shall toggle to the “Sign-Out” button to provide the user with a safe and quick way to log out of the system once the work is done.

REQ-1.5: The UMGC City Admin Portal system shall provide a user with a web page dedicated to provide help and support by clicking on the “Help” button. Here the user shall be presented with ways to receive technical support. In addition, the user can self-serve and retrieve his/her log-in account credentials (i.e., email address and/or password).

REQ-1.6: The user shall be brought to the landing page once successfully logged into the system. The system features of creating a new use case, editing existing use cases, and uploading a use case are now available. The user can navigate from anywhere within the app and back to the landing page by clicking on the “Home Link” logo. However, the user must continue to be logged in for this; otherwise, the user shall be brought to the home page instead.

REQ-1.7: The UMGC City Admin Portal system works by accepting data inputted by the user. The user creates a use case for his/her city. The application shall map the use case to the appropriate city zoning code(s), definition(s), regulation(s), ordinance(s), permit requirement(s), application(s), etc., populating database for output.

REQ-1.8: The UMGC City Admin Portal system shall allow a user to manually add a new use case, edit existing use cases, and delete a use case from a list within the account. In addition, the system shall allow the user to upload use cases from an external file using the “Upload Use Case” button. The system shall validate the file located in the user specified directory that it follows specific formats and standards. If acceptable, the system shall automatically populate proper data fields, create use cases, and add them to a list of use cases within the account.

REQ-1.9: The UMGC City Admin Portal system shall provide a user with a certain amount of memory storage space that is linked to his/her account. The user can save the use cases and come back to them later. Once the storage runs out, the user must delete existing use case(s) to make room for more space or contact technical support for additional storage.

REQ-1.10: The UMGC City Admin Portal system shall process the project data inputted by the user and generate proper output of static HTML page with appropriate DDL scripts using the “Generate Output” button for exporting the project. Once the output is generated, the functions of the ‘Preview’ button and the “Download” button become active. The user is presented with a display preview by clicking on the “Preview” button. Here, the user can make sure that the HTML page has the correct look and feel. Once satisfied, the user can click on the “Download” button to save the output file of project export data to be used to improve the existing city website.

### 3.2.2 Map Function for City Resident Users

The primary interface for this application includes an interactive map.

### 3.2.2.1 Description and Priority

Each zone on the map is clickable. When a user clicks on a zone, a modal opens to display all applicable information for the specific zone. Priority = High.



### 3.2.2.2 Stimulus/Response Sequences

Stimulus: The user drags either scrollbar.

Response: The city map scrolls horizontally or vertically.

Stimulus: The user clicks on a zone within the map.

Response: A modal displays specific information pertaining to the selected zone.

Stimulus: The user clicks on the “Zone Information” link provided for the zone.

Response: The browser navigates to an external page in a separate window to display relevant zone information. The applicable link is retrieved from the city database.

Stimulus: The user clicks on any of the “Application” links provided for the zone.

Response: The browser navigates to an external page in a separate window to display relevant applications for use cases allowed within the zone. The applicable links are retrieved from the city database.

Stimulus: The user clicks on the “X” button located on the top right of the modal.

Response: The modal closes and the view is returned to the map page.

Stimulus: The user clicks on the “Quick Reference Page” link.

Response: The browser navigates to the “Quick Reference Page” in a separate window.

Stimulus: The user clicks on the “Need Help?” icon located on the bottom right of the page.

Response: The ChatBot application opens on top of the map interface page (implemented by UMGC City Team 2).

### 3.2.2.3 Functional Requirements

REQ-2.1: When the user drags either scrollbar, the map image is moved within the containing window. Scrolling is set to default Chrome settings.

REQ-2.2: When the user clicks on a zone within the map, a modal opens on top of the map to display specific information to the selected zone. The modal is designed using in-line JavaScript and custom CSS.

REQ-2.3: When the user clicks on the “Zone Information” link, the browser navigates to an external page on the Municode website that contains information that is relevant to the selected zone. The UMGC City Teams are not responsible for the accuracy or validity of the content found on this external site.

REQ-2.4: When the user clicks on any “Application” link, the browser navigates to an external page on the Municode website that contains the applications that are relevant to the selected zone. The UMGC City Teams are not responsible for the accuracy or validity of the content found on this external site.

REQ-2.5: When the user clicks on the “Quick Reference Page” link, the browser navigates to an internal page that contains information designed by the city officials. The information on this page is retrieved from the database. The UMGC City Teams are not responsible for the accuracy or validity of the content found on any external site.

REQ-2.6: When the user clicks on the “Need Help?” icon, the ChatBot application opens on top of the map interface. The ChatBot application is designed by UMGC City Team 2 and is not addressed in further detail in this document.

### 3.2.3 Quick Reference Page (Land Uses)

The “Quick Reference Page” allows the user to view all the land use cases and easily access specific regulations and applications.

### 3.2.3.1 Description and Priority

The Quick Reference Page displays different allowed land usages as defined by the city representative. Under each land use option, there will be links to regulations and applications for those specific types. The links will take the user directly to the appropriate regulation or application. Priority = High

### 3.2.3.2 Stimulus/Response Sequences

Stimulus: The user clicks on the “Quick Reference Page” link on the main page.

Response: The system directs the user to the “Quick Reference Page” which will show the user all the different land uses, regulation and applications.

### 3.2.3.3 Functional Requirements

REQ-3.1: The system shall direct the user to the “Quick Reference Page”.

### 3.2.4 Home Occupation Permit

The “Home Occupation Permit” use case was defined by the client and will be included in the initial delivery. The requirements specifically related to this use case are defined in the next three subsections.

### 3.2.4.1 Description and Priority

The “Home Occupation Permit” option allows the user to retrieve regulations and provides a link to the application. Priority = High

#### Stimulus/Response Sequences

Stimulus: The user clicks on the link to the regulations for “Home Occupation Permit”.

Response: The system directs the user to the specific regulation.

Stimulus: The user clicks on the link to the applications for “Home Occupation Permit”.

Response: The system directs the user to the specific application.

### 3.2.4.3 Functional Requirements

REQ-4.1: The system shall direct the user to the “Home Occupation” regulations page after the user clicks the regulation link.

REQ-4.2: The system shall direct the user to the “Home Occupation” application page after the user clicks the application link.

### 3.2.5 Accessory Dwelling Unit

The “Accessory Dwelling Unit” use case was defined by the client and will be included in the initial delivery. The requirements specifically related to this use case are defined in the next three subsections.

#### Description and Priority

The “Accessory Dwelling Unit” option allows the user to retrieve regulations and provides a link to application. Priority = High.

### 3.2.5.2 Stimulus/Response Sequences

Stimulus: The user clicks on the link to the regulations for “Accessory Dwelling Unit”.

Response: The system directs the user to the specific regulation.

Stimulus: The user clicks on the link to the applications for “Accessory Dwelling Unit”.

Response: The system directs the user to the specific application.

### 3.2.5.3 Functional Requirements

REQ-5.1: The system shall direct the user to the “Accessory Dwelling Unit” regulations page after the user clicks the regulations link.

REQ-5.2: The system shall direct the user to the “Accessory Dwelling Unit” application page after the user clicks the application link.

### 3.2.6 Short Term Rental

The “Short Term Rental” use case was defined by the client and will be included in the initial delivery. The requirements specifically related to this use case are defined in the next three subsections.

#### Description and Priority

The “Short Term Rental” option allows the user to retrieve regulations and provides a link to application. Priority = High.

#### Stimulus/Response Sequences

Stimulus: The user clicks on the link to the regulations for “Short Term Rental”.

Response: The system directs the user to the specific regulation.

Stimulus: The user clicks on the link to the applications for “Short Term Rental”.

Response: The system directs the user to the specific application.

#### Functional Requirements

REQ-6.1: The system shall direct the user to the “Short Term Rental” regulations page after the user clicks the regulations link.

REQ-6.2: The system shall direct the user to the “Short Term Rental” application page after the user clicks the application link.

### 3.2.7 Accessory Structure

The “Accessory Structure” use case was defined by the client and will be included in the initial delivery. The requirements specifically related to this use case are defined in the next three subsections.

### 3.2.7.1 Description and Priority

The “Accessory Structure” option allows the user to retrieve regulations and provides a link to

the application. Priority = High.

#### Stimulus/Response Sequences

Stimulus: The user clicks on the link to the regulations for “Accessory Structure”.

Response: The system directs the user to the specific regulation.

Stimulus: The user clicks on the link to the applications for “Accessory Structure”.

Response: The system directs the user to the specific application.

### 3.2.7.3 Functional Requirements

REQ-7.1: The system shall direct the user to the “Accessory Structure” regulations page after the user clicks the regulations link.

REQ-7.2: The system shall direct the user to the “Accessory Structure” application page after the user clicks the application link.

### 3.2.8 Exterior Modifications in Landmark District The “Exterior Modifications in Landmark District” use case was defined by the client and will be included in the initial delivery. The requirements specifically related to this use case are defined in the next three subsections.

#### 3.2.8.1 Description and Priority

The “Exterior Modifications in Landmark District” option allows the user to retrieve regulations and provides a link to application. Priority = High.

#### 3.2.8.2 Stimulus/Response Sequences

Stimulus: The user clicks on the link(s) to the regulation(s) for “Exterior Modifications in Landmark District”.

Response: The system directs the user to the specific regulation.

Stimulus: The user clicks on the link to the applications for “Exterior Modifications in Landmark District”.

Response: The system directs the user to the specific application.

#### 3.2.8.3 Functional Requirements

REQ-8.1: The system shall direct the user to the “Exterior Modifications in Landmark District” regulations page after the user clicks the regulations link.

REQ-8.2: The system shall direct the user to the “Exterior Modifications in Landmark District” application page after the user clicks the application link.

### 3.2.9 Fence

The “Fence” use case was defined by the client and will be included in the initial delivery. The requirements specifically related to this use case are defined in the next three subsections.

### 3.2.9.1 Description and Priority

The “Fence” option allows the user to retrieve regulations and provides a link to application. Priority = High.

#### Stimulus/Response Sequences

Stimulus: The user clicks on the link to the regulations for “Fence”.

Response: The system directs the user to the specific regulation.

Stimulus: The user clicks on the link to the applications for “Fence”.

Response: The system directs the user to the specific application.

#### Functional Requirements

REQ-9.1: The system shall direct the user to the “Fence” regulations page after the user clicks the regulations link.

REQ-9.2: The system shall direct the user to the “Fence” application page after the user clicks the application link.

### 3.2.10 Tree Removal (Residential) The “Tree Removal (Residential)” use case was defined by the client and will be included in the initial delivery. The requirements specifically related to this use case are defined in the next three subsections.

### 3.2.10.1 Description and Priority

The “Tree Removal (Residential)” option allows the user to retrieve regulations and provides a link to application. Priority = High.

### 3.2.10.2 Stimulus/Response Sequences

Stimulus: The user clicks on the link to the regulations for “Tree Removal (Residential)”.

Response: The system directs the user to the specific regulation.

Stimulus: The user clicks on the link to the applications for “Tree Removal (Residential)”.

Response: The system directs the user to the specific application.

### 3.2.10.3 Functional Requirements

REQ-10.1: The system shall direct the user to the “Tree Removal (Residential)” regulations page after the user clicks the regulations link.

REQ-10.2: The system shall direct the user to the “Tree Removal (Residential)” application page after the user clicks the application link.

### 3.2.11 Day-Care The “Day-Care” use case was defined by the client and will be included in the initial delivery. The requirements specifically related to this use case are defined in the next three subsections.

#### Description and Priority

The “Day-Care” option allows the user to retrieve regulations and provides a link to application. Priority = High.

#### Stimulus/Response Sequences

Stimulus: The user clicks on the link to the regulations for “Day-Care”.

Response: The system directs the user to the specific regulation.

Stimulus: The user clicks on the link to the applications for “Day-Care”.

Response: The system directs the user to the specific application.

### Functional Requirements

REQ-11.1: The system shall direct the user to the “Day-Care” regulations page after the user clicks the regulations link.

REQ-11.2: The system shall direct the user to the “Day-Care” application page after the user clicks the application link.

### 3.2.12 Sober Living Facility The “Sober Living Facility” use case was defined by the client and will be included in the initial delivery. The requirements specifically related to this use case are defined in the next three subsections.

### 3.2.12.1 Description and Priority

The “Sober Living Facility” option allows the user to retrieve regulations and provides a link to

the application. Priority = High.

#### Stimulus/Response Sequences

Stimulus: The user clicks on the link to the regulations for “Sober Living Facility”.

Response: The system directs the user to the specific regulation.

Stimulus: The user clicks on the link to the applications for “Sober Living Facility”.

Response: The system directs the user to the specific application.

### 3.2.12.3 Functional Requirements

REQ-12.1: The system shall direct the user to the “Sober Living Facility” regulations page after the user clicks the regulations link.

REQ-12.2: The system shall direct the user to the “Sober Living Facility” application page after the user clicks the application link.

## 3.3 Design and Implementation Constraints

* The system will use the Model-View-Controller design pattern to segregate the business logic from the user interface templates.
* The application back-end programming will be written in Java. Java SE 11 will be used due to its planned long-term support.
* The application will use the Spring Boot utility tool for implementing a RESTful backend microservice.
* The application front-end scripts will be written in JavaScript and Angular, a TypeScript-based web application framework.
* The application will log application errors and audit trails using Log4j, a Java-based logging library.
* The application’s dependencies will be managed using Maven, a build automation tool for Java projects.
* The application code will include SQL commands to query the database. The application will not use an Object-Relational Mapper (ORM).
* The application code will be managed using Git, a distributed source-control system. The Git repository will be hosted in GitHub.
* The application will be accessible via the internet. The application will not be available if internet connection is lost between the client, the application server, or the database server.
* The application will not offer the same experience when displayed on a small screen such as a tablet or mobile phone.
* The application performance may be constrained by the server’s hardware capacity.

## 3.4 User Documentation

The final product will be accompanied by the following documentation:

* A copy of the approved final project plan
* A copy of the approved final Software Requirements Specification (SRS)
* A copy of the Software Test Plan with test results
* Interface Design Documentation (Tutorial)

# Nonfunctional Requirements

The following subsections provide an in-depth view of the nonfunctional requirements related to the UMGC City Application. Nonfunctional requirements specify the performance qualities that the application must possess.

## 4.1 Performance

NF-1.1: The system shall support 300 simultaneous users from 8am-8pm (local time); 150 simultaneous users at all other times.

NF-1.2: All web pages generated by the system shall be generated and displayable within 5 seconds, using picture placeholders until pictures are available. (“IMG” image tags shall use the “ALT” field.)

NF-1.3: The application must respond to user input within 2 seconds.

NF-1.4: The system shall download new status parameters within 5 minutes of a change.

NF-1.5: The system shall be available for use no less than 351 days per year (99% availability).

## 4.2 Security

NF-2.1: The system shall permit only city users with an authorized application account to create, modify, or delete data related to their own account.

NF-2.2: The system shall have a time out feature of 5 minutes to protect user’s privacy.

## 4.3 Quality

NF-3.1: The features must provide feedback to the user within five seconds on all systems that possess the following specifications (at a minimum):

* Computer and processor: 1 gigahertz (GHz) or faster x86-bit or x64-bit processor with SSE2 instruction set
* Memory: 2 GB RAM
* Hard disk: 3.0 GB of available disk space
* Display PC: 1024 x 768 screen resolution
* Graphics: hardware acceleration requires a DirectX 10 graphics card.
* Operating system: Windows 10, Windows 8.1, Windows 8, Windows 7 Service Pack 1, Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, or Windows Server 2008 R2
* Google Chrome, version 80 or later

NF-3.2: The system shall be available to users 99.9% of the time.

## 4.4 Cultural, Political and Accessibility

NF-4.1: Personal information is protected in compliance with the Data Protection Act.

NF-4.2: The system shall provide individuals with disabilities with the information and data involved by an alternative means of access that allows the individual to use the information and data.

NF-4.3: The system shall have an equivalent alternative for any multimedia presentation that can be synchronized with all presentation.

NF-4.4: The system shall be designed so that all information conveyed with color is also available without color, for example from context or markup.

NF-4.5: The system documents shall be organized so they are readable without requiring an associated style sheet.

NF-4.6: The system shall have redundant text links which will be provided for each active region of a server-side image map.

NF-4.7: The system shall use images that are used to identify controls, status indicators, or other programmatic elements. The meaning assigned to those images shall be consistent throughout the application’s performance.

NF-4.8: The system shall not override user selected contrast and color selections and other individual display attributes.

NF-4.9: The system shall have a text equivalent for every non-text element that is provided (e.g., via “alt”, “longdesc”, or in element content).

NF-4.10: The system pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.

NF-4.11: The system frames shall be titled with text that facilitates frame identification and navigation.

NF-4.12: The system Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.

### Appendix A

### User Stories

Table 4 - Agile User Stories

| **Story #** | **As a/an** | **I want to…** | **So that…** | **Priority** | **Acceptance Criteria** | **SRS Reference** |
| --- | --- | --- | --- | --- | --- | --- |
| 001 | City resident | use a map to find my zone | I can quickly find what zone(s) I live in. | High | When a city resident user clicks on a zone in the map interface, a modal displays the zone name(s) that apply to the selected zone. | Section [3.2.2](#_3.2.2__) |
| 002 | City resident | see only the land use cases that pertain to my zone/property | I can quickly find the information I need, without crawling through an entire municipal code. | High | When a city resident user clicks on a zone in the map interface, a modal displays the links to all the regulation(s) that apply to the selected zone. | Sections:  [3.2.1](#_3.2.1__)  [3.2.2](#_3.2.2__)  [3.2.3](#_3.2.3__)  [3.2.4](#_3.2.4__)  [3.2.5](#_3.2.5__)  [3.2.6](#_3.2.6__)  [3.2.7](#_3.2.7__)  [3.2.8](#_Exterior_Modifications_in)  [3.2.9](#_3.2.9__)  [3.2.10](#_3.2.10__)  [3.2.11](#_3.2.11__)  [3.2.12](#_3.2.12__) |
| 003 | City resident | see applications for only the land use cases that pertain to my zone/property | I can quickly find the applications I need, without crawling through an entire municipal code. | High | When a city resident user clicks on a zone in the map interface, a modal displays the links to all the application(s) that apply to the selected zone. | Sections:  [3.2.1](#_3.2.1__)  [3.2.2](#_3.2.2__)  [3.2.3](#_3.2.3__)  [3.2.4](#_3.2.4__)  [3.2.5](#_3.2.5__)  [3.2.6](#_3.2.6__)  [3.2.7](#_3.2.7__)  [3.2.8](#_Exterior_Modifications_in)  [3.2.9](#_3.2.9__)  [3.2.10](#_3.2.10__)  [3.2.11](#_3.2.11__)  [3.2.12](#_3.2.12__) |
| 004 | City management | present a quick reference page | our city residents can quickly find information on popular zoning topics. | High | When a city management user selects to create the quick reference page, the application creates a customized HTML page that includes all use cases saved in the city’s database | Section [3.2.3](#_3.2.3__) |
| 005 | City management | create a user-friendly map interface | our city residents can quickly find accurate information regarding the zone(s) they live in. | High | [Only applies to the pilot client, “The City of Pasadena”] . A map will be displayed on the welcome page to the UMGC City Application. When a city resident user clicks on a zone in the map interface, a modal displays the zone name(s), and links to regulation(s) and application(s) that apply to the selected zone. | Section [3.2.2](#_3.2.2__) |
| 006 | City management user | create a customized database for the city zoning laws that are most frequently searched | our city residents can rely on our website services and our in-office staff can attend to licensing matters in a quicker fashion. | High | The UMGC City Application presents an interface for the City Management User to add information to the database. The user should be able to confirm that Add/Delete/Update statements have been applied. The output Quick Reference HTML page should contain all the information added to the database by the client. | Section [3.2.1](#_3.2.1__) |