CMSC 447

Software Development Plan (SDP)

# Scope

This Software Development Plan (SDP) establishes the plan for software implementation, test, and qualification for Project. The Project is being developed under the direction of the Team 1. Updates to this SDP will address future Project software changes.

## Identification

The Project consists of the 4 software CSCIs listed below:

1. (DATA) Data collector: Responsible for communicating between API’s.
2. (IMD) Interactive Map Display: Responsible for displaying a map to the client-side system.
3. (FORM) Search Form: Responsible for aggregating the information given by the user containing filters and a location.
4. (WEB) Responsible for serving HTML from the server-side system to the client-side system and hosting an API to which the (FORM) sends the filters and location.

## System overview

The system will be a web-based application with the ability to search for houses based off of certain search criterias in a census designated area from a map interface. The user will be any individual searching for a home in the region from Ocean Pines to the coast of Delaware. There be unit tests after each feature is added. The operating site will be the RLC at UMBC.

## Referenced documents

This section shall list the number, title, revision, and date of all documents referenced in this plan.

* Software Requirement Specification
* Software Design Document
* Software Test Description
* Software Test Report
* Software User Manual

# Overview of required work

This section shall be divided into paragraphs as needed to establish the context for the planning described in later sections. It shall include, as applicable, an overview of:

* Requirements and constraints on the system and software to be developed
  + System Requirements
* Requirements and constraints on project documentation
  + Refer to Standards for Software Products
* Requirements and constraints on project schedules and resources
  + Project due 05/10/2018

# Plans for performing general software development activities

This section shall be divided into the following paragraphs. Provisions corresponding to non-required activities may be satisfied by the words "Not applicable." If different builds or different software on the project require different planning, these differences shall be noted in the paragraphs. In addition to the content specified below, each paragraph shall identify applicable risks/uncertainties and plans for dealing with them.

## Software development process

This project will be implemented with an object oriented design approach and we will map classes using class diagrams. We will utilize API’s from Google maps, Zillow, and the Census Bureau to gather information to generate property listings. The requests to obtain the data shall be validated by unit testing.

## General plans for software development

Project software development will conform to standards stated below in 3.2. Team 1 has tailored these standards, practices, and processes to Project software development, as described in Section 4 of this SDP.

### Software development methods

The project will gather functional requirements of what the application will do, and implement the requirements in the software application. design, coding, and testing will conform to the SRS and SDP.

### Standards for software products

Team 1 will comply with applicable directions listed below. These standards that are applicable to software requirements, design, coding, testing, and data.

* + - 1. Standards for format (such as indentation, spacing, capitalization, and order of information)
* Indentation - 4 spaces
  + - 1. Standards for header comments
* File Headers
* Function Headers- excluding setters and getters
  + - 1. Standards for other comments
* Comments other than header comments shall be in whatever format the writer chooses
  + - 1. Naming conventions for variables, parameters, packages, procedures, files, etc.
* The first letter of class names will be uppercase
* Functions will be lowercase/camelcase
* Variables will be lowercase/camelcase
* Constants will be all uppercase with underscores for spaces

* + - 1. Restrictions, if any, on the complexity of code aggregates
* No tertiary structures

# Plans for performing detailed software development activities

The Waterfall Life Cycle Model will be used to guide the content and format developing.

## Project planning and oversight

This SDP shall be maintained and modified to reflect the current plans, policies, processes,  
resources, and standards affecting the Project.

### System test planning

The intent of system test planning is to validate that the system meets its performance requirements. It will be the responsibility of the Team1 to direct the development of system test plans and procedures

## System requirements analysis

Team 1 will process requests for clarification, change, and waiver/deviation.

### Analysis of user input

Any questions and concerns given by the user(s) shall be answered with feedback from a member of Team 1. Documentation of all discussions of Team 1 and the user(s) will be kept.

## Software design

This paragraph shall be divided into the following subparagraphs to describe the approach to be followed for software design. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic. (insert design picture/DIAGRAM)

## Software implementation and unit testing

The purpose of Software Implementation and Unit Testing is to implement and test the detailed design for the Project. The following paragraphs describe Software Implementation and Unit Testing processes.

### Software implementation

The purpose of software implementation and unit testing is to create qualification test ready CSCI functional components by implementing and testing each components software unit identified in the detailed design. This may involve developing new code and/or modifications of existing code, and following documented programming style guidelines.

### Preparing for unit testing

Make sure all the packages and other files are present before testing.

### Performing unit testing

Run unit tests on all unit testable pieces of code. This includes all functions written, even setters and getters.

### Revision and retesting

If any test fails and the reason is unknown, then alert the person responsible for the feature.

## Unit integration and testing

The purpose of integration and test is to incrementally integrate software units into larger software components, and components into a complete system. Testing is performed to validate each component's ability to meet its stated requirements and to ensure interoperability of the major software components. Integration continues until all software components are integrated with the system-level hardware suite into a single functioning system.

### Preparing for unit integration and testing

Ensure the software needed for integration testing is running and working.

### Performing unit integration and testing

Run all integration tests twice and examine the state of the system, make sure are.

### Revision and retesting

If any test fails and the reason is unknown, then alert the person responsible for the feature.

## Preparing for software use

Team 1 is responsible for correctly packaging the software.

### Preparing the executable software

The executable software will be prepared with any batch files, command files, data files, or other software files.

### Preparing user manuals

User Manuals will be prepared and validated by Team 1.

## Software quality assurance

Software quality assurance ensures that discovered defects are corrected and open issues are resolved.

### Software quality assurance evaluations

Software quality assurance will be evaluated by if map points of homes are displayed correctly from unit testing.

### Software quality assurance records, including items to be recorded

## Schedules and activity network

This section shall present:

1. Schedule(s) identifying the activities in each build and showing initiation of each activity, availability of draft and final deliverables and other milestones, and completion of each activity
2. An activity network, depicting sequential relationships and dependencies among activities and identifying those activities that impose the greatest time restrictions on the project

## Notes

* API: Application Programming Interface.
* CDP: Census Designated Place, any place for which census data is available.
* CSCI: Computer Software Configuration Item.
* RLC: Retriever Learning Center.
* SDP: Software Development Plan.
* SRS: Software Requirement Specification.
* UMBC: University of Maryland Baltimore County