Software Requirements Specification Template

CS 353—Software Engineering

Spring 2020

The following annotated template shall be used to complete the Software Requirements Specification (SRS) assignment of CS 353. The instructor must approve any modifications to the overall structure of this document.

**Template Usage:**

Text contained within angle brackets (‘<’, ‘>’) shall be replaced by your project-specific information and/or details. For example, <Project Name> will be replaced with either ‘Smart Home’ or ‘Sensor Network’.

Italicized text is included to briefly annotate the purpose of each section within this template. This text should not appear in the final version of your submitted SRS.

This cover page is not a part of the final template and should be removed before your SRS is submitted.

**Acknowledgements:**

Sections of this document are based upon the IEEE Guide to Software Requirements Specification (ANSI/IEEE Std. 830-1984).

Scrap-Bot

Software Requirements Specification

Version 1.0

2 – 4 – 2020

Hasan Haider

Inara Chagani

Zainab Ali

Raza Naqvi  
Muhammad Shahzain

Software Engineers

Prepared for

CS 353—Software Engineering

Instructor: Umair Azfar Khan.

Spring 2020

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| <3/3/2020> | Version 1.0 | Muhammad Shahzain | <First Revision> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | <Your Name> |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

**REVISION HISTORY II**

**DOCUMENT APPROVAL II**

**1. INTRODUCTION 1**

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 1

1.5 Overview 1

**2. GENERAL DESCRIPTION 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 General Constraints 2

2.5 Assumptions and Dependencies 2

**3. SPECIFIC REQUIREMENTS 2**

3.1 External Interface Requirements 3

*3.1.1 User Interfaces 3*

*3.1.2 Hardware Interfaces 3*

*3.1.3 Software Interfaces 3*

*3.1.4 Communications Interfaces 3*

3.2 Functional Requirements 3

*3.2.1 <Functional Requirement or Feature #1> 3*

*3.2.2 <Functional Requirement or Feature #2> 3*

3.3 Use Cases 3

*3.3.1 Use Case #1 3*

*3.3.2 Use Case #2 3*

3.4 Classes / Objects 3

*3.4.1 <Class / Object #1> 3*

*3.4.2 <Class / Object #2> 3*

3.5 Non-Functional Requirements 4

*3.5.1 Performance 4*

*3.5.2 Reliability 4*

*3.5.3 Availability 4*

*3.5.4 Security 4*

*3.5.5 Maintainability 4*

*3.5.6 Portability 4*

3.6 Inverse Requirements 4

3.7 Design Constraints 4

3.8 Logical Database Requirements 4

3.9 Other Requirements 4

**4. ANALYSIS MODELS 4**

4.1 Sequence Diagrams 5

4.3 Data Flow Diagrams (DFD) 5

4.2 State-Transition Diagrams (STD) 5

**5. CHANGE MANAGEMENT PROCESS 5**

**A. APPENDICES 5**

A.1 Appendix 1 5

A.2 Appendix 2 5

# Introduction

Scrap-Bot is an automated ad-fetching and posting based website that primarily uses the concepts of web-crawling and web-scraping to get advertisements of automobiles to our website.  
The user can set preferences of what types of automobile advertisements they want as what appeals to their interest of buying or selling. This document will serve as a set of major guidelines required to design and develop the Scrap-Bot. It also shows the complete purpose of development of the system, and will contain interfaces, system constraints required by the intended user (Talha Amin).

*The introduction to the Software Requirement Specification (SRS) document should provide an overview of the complete SRS document. While writing this document please remember that this document should contain all of the information needed by a software engineer to adequately design and implement the software product described by the requirements listed in this document. (Note: the following subsection annotates are largely taken from the IEEE Guide to SRS).*

## 1.1 Purpose

*What is the purpose of this SRS and the (intended) audience for which it is written?*

The purpose of this piece of writing is to document all the functional and non-functional requirements that are to be expected in the Scrap-Bot end-product. The intended audience are the software developers, programmers, designers, testers and project managers.

## 1.2 Scope

The document is based on the Scrap-Bot which is a web-based application that fetches advertisements from automobile buy-sell websites and creates a friendly ecosystem for the user according to what preferences they have set.

It will provide a one-stop platform to those who are looking for a specific automobile but don’t want to search through all different websites.

*This subsection should:*

*(1) Identify the software product(s) to be produced by name; for example, Host DBMS, Report Generator, etc*

*(2) Explain what the software product(s) will, and, if necessary, will not do*

*(3) Describe the application of the software being specified. As a portion of this, it should:*

*(a) Describe all relevant benefits, objectives, and goals as precisely as possible. For example, to say that one goal is to provide effective reporting capabilities is not as good as saying parameter-driven, user-definable reports with a 2 h turnaround and on-line entry of user parameters.*

*(b) Be consistent with similar statements in higher-level specifications (for example, the System Requirement Specification), if they exist. What is the scope of this software product.*

## 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Project-Manager** | **One who handles the planning and execution of a certain project** |
| **Web-crawling** | **A technique mostly used by search engines to systematically browse the WWW(Internet)** |
| **WWW** | **WorldWideWeb** |
| **Web Scraping** | **A process in which large amounts of data are extracted from the web and is usually saved in a database.** |
| **Database** | **A systematic way of organizing data to ensure fast access, insertion and deletion of data.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

*This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.*

## 1.4 References

*This subsection should:*

*(1) Provide a complete list of all documents referenced elsewhere in the SRS.*

*(2) Specify the sources from which the references can be obtained.*

*This information may be provided by reference to an appendix.*

## 1.5 Overview

*This subsection should:*

*(1) Describe what the rest of the SRS contains*

*(2) Explain how the SRS is organized.*

The SRS is divided into five components. Beginning with the introduction, it jumps onto the

General Description, then comes onto Requirements followed by Analysis Models and

Appendix.

The General Description has all the specifications of the product in a high level fashion while the

Requirements contains detailed description of what the product should be able to do, or contain.

Analysis Models tell us the description of certain requirements while the appendices contain all

the additional information related to SRS.

# 2. General Description

*This section of the SRS should describe the general factors that affect 'the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.*

## 2.1 Product Perspective

This product is being developed as an innovation to basic primary buy-sell websites like OLX or PakWheels, (incomplete)

*This subsection of the SRS puts the product into perspective with other related products or projects.*

## 2.2 Product Functions

*This subsection of the SRS should provide a summary of the functions that the software will perform.*

Log In

Log Out

Sign Up

Delete Account

Set user Preferences

Post Ads-Create Ads

Post Ads- Delete Ads

Post Ads- Update Ads

Create User Wishlist

## 2.3 User Characteristics

*This subsection of the SRS should describe those general characteristics of the eventual users of the product that will affect the specific requirements.*

## 2.4 General Constraints

*This subsection of the SRS should provide a general description of any other items that will*

*limit the developer’s options for designing the system.*

## 2.5 Assumptions and Dependencies

*This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.*

# 3. Requirements

*This will be the largest and most important section of the SRS.*

*Each requirement in this section should be:*

* *Correct*
* *Traceable (both forward and backward to prior/future artifacts)*
* *Unambiguous*
* *Verifiable (i.e., testable)*
* *Prioritized (with respect to importance and/or stability)*
* *Complete*
* *Consistent*
* *Uniquely identifiable (usually via numbering like 3.4.5.6)*

*Attention should be paid to the carefully organize the requirements presented in this section so that they may easily accessed and understood.*

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

*Is it for mobile, web, or standalone client? What will be the interface for each?*

It will have a web-application based interface.

### 3.1.2 Hardware Interfaces

*On what hardware will it be implemented?*

Hardware specifics are not related to standalone devices. The main system will have a hosted server while the user system can be a simple personal computer.

### 3.1.3 Software Interfaces ( X )

*Is there requirement for a specific software? If there is an external software, what is it?*

### 3.1.4 Communications Interfaces

*Is it web/LAN dependent? Or any other communication service?*

It’s mostly web dependent that means it must be hosted on a fast-response server for efficient retrieval of data.

## 3.2 Functional Requirements

*This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.*

### 3.2.1 Sign Up and Log In

3.2.1.1 Introduction

The user should be able to sign up on the platform to use it.

3.2.1.2 Inputs

The user will connect to the website and click on the Sign Up button.

After signing up, the user will click the log in button where the user inputs the credentials.

3.2.1.3 Processing

The user credentials will be saved in the credential database when sigining up

And these credentials will be verified upon logging in

3.2.1.4 Outputs

The user will be signed up OR logged in, and will be able to use the system.

3.2.1.5 Error Handling

If user is already registered, signing up will return an error.

If user inputs wrong credentials, log in won’t be allowed

3.2.2 **Set Preferences**

3.2.2.1 **Introduction**

The user should be able to set preferences about the make and type of automobile they want to see.

3.2.2.2 **Input**

The user will select “Preferences” button where the user will have the option to type and save the preferences.

3.2.2.3 **Processing**

The settings will be saved in the Scrap-Bot Database.

3.2.2.4 **Output**

The ad-fetching mechanism will work based on these preferences now.

### 3.2.3 Fetch Advertisements

3.2.3.1 **Introduction**

The user should be able to fetch ads upon the basis of the preferences they have set

3.2.3.2 **Inputs**

There will be no user inputs in this case as the Scrap-Bot will fetch ads on the basis of preferences that are set.

3.2.3.3 **Processing**

The Scrap-Bot will fetch user preferences from the database and will start the process of web-crawling to get relevant advertisements.

3.2.4.4 **Outputs**

The user will start receiving relative ads on the Scrap-Bot platform.

…

## 3.3 Use Cases

# 3.3.1 Use Case #1 Create Advertisement

# Brief Description

This use case describes how the member uses the website to post an advertisement from his/her account.

**Actors**

**Member**

**Scrap-Bot Database**

**Preconditions**

There is an active network connection to the website.

The User already has an account on the website.

The User has already logged-in through a valid id and password.

**Basic Flow of Events**

1. The use case begins when User clicks on the “Post your Ad” button.
2. Use Case: Create Advertisement is performed.
3. The system prompts for car details.
4. The system then asks the user to select the websites he/she wants to post the advertisement on.
5. The user may then update previous credentials for the websites.
6. The advertisement is posted.
7. The use case ends successfully

**Alternative Flows**

**Incomplete Car Details**

If in step 3 of the basic flow Member the use case: take Credentials does not complete successfully, then

1. The user is prompted to fill in complete information.
2. The use case resumes at step 3.

**Websites not selected**

If in step 4 of the basic flow the member does not select any website, then

1. The website will select Scrap-Bot to post the advertisement on.
2. The use case resumes at step 5.

**Quit**

If at point prior to step 7 in the basic flow the Member selects Quit, then

1. The system prompts the user that all activities have been cancelled.
2. The use case ends

**Key Scenarios**

**No Response from Website**

**Post-conditions**

**Successful Completion**

The Member has successfully posted the advertisement and the database has been updated.

**Failure Condition**

The logs have been updated accordingly.

**Special Requirements**

All steps are performed in binary. That is they are either performed successfully or they fail entirely.  
The Database is not updated until the advertisement is posted.

# 3.3.2 Use Case #2 – Delete Advertisement

**Brief Description**

This use case describes how the member uses the website to delete an advertisement from his/her account.

**Actors**

**Member**

**Scrap-Bot Database**

**Preconditions**

There is an active network connection to the website.

The User already has an account on the website.

The User has already logged-in through a valid id and password.

The User has already created and posted an advertisement.

**Basic Flow of Events**

1. The use case begins when User clicks on the “Delete your Ad” button.
2. Use Case: Delete Advertisement is performed.
3. The system removes advertisement details from the database.
4. The use case ends successfully

**Alternative Flows**

**Quit**

If at point prior to step 4 in the basic flow the Member selects Quit, then

1. The system prompts the user that all activities have been cancelled.
2. The use case ends

**Key Scenarios**

**No Response from Website**

**Post-conditions**

**Successful Completion**

The Member has successfully deleted the advertisement and the database has been updated.

**Failure Condition**

The logs have been updated accordingly.

**Special Requirements**

All steps are performed in binary. That is they are either performed successfully or they fail entirely.  
The Database is not updated until the advertisement is deleted.

# 3.3.3 Use Case #3-Update Advertisement

# Brief Description

This use case describes how the member uses the website to update an advertisement from his/her account.

# Actors

## Member

## Scrap-Bot Database

# Preconditions

There is an active network connection to the website.

The User already has an account on the website.

The User has already logged-in through a valid id and password.

The User has already created and posted the advertisement.

# Basic Flow of Events

1. The use case begins when User clicks on the “Update your Ad” button.
2. Use Case: Update Advertisement is performed.
3. The system prompts for car details.
4. The system then asks the user to select the websites he/she wants to post the advertisement on.
5. The user may then update previous credentials for the websites.
6. The advertisement is posted.
7. The use case ends successfully

# Alternative Flows

## Incomplete Car Details

If in step 3 of the basic flow Member the use case: take Credentials does not complete successfully, then

1. The user is prompted to fill in complete information.
2. The use case resumes at step 3.

## Websites not selected

If in step 4 of the basic flow the member does not select any website, then

1. The website will select Scrap-Bot to post the advertisement on.
2. The use case resumes at step 5.

## Quit

If at point prior to step 7 in the basic flow the Member selects Quit or logs out, then

1. The system prompts the user that all activities have been cancelled.
2. The use case ends

## No new changes

If the User does not make any new changes, then

1. The database is not updated.
2. Use case resumes at step 5.

# Key Scenarios

## No Response from Website

# Post-conditions

## Successful Completion

The Member has successfully posted the advertisement and the database has been updated.

## Failure Condition

The logs have been updated accordingly.

# Special Requirements

All steps are performed in binary. That is they are either performed successfully or they fail entirely.  
The Database is not updated until the advertisement is posted.

## 3.4 Classes / Objects <Should be completed in Assignment #3>

### 3.4.1 <Class / Object #1>

3.4.1.1 Attributes

3.4.1.2 Functions

<Reference to functional requirements and/or use cases>

### 3.4.2 <Class / Object #2>

…

## 3.5 Non-Functional Requirements

*Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).*

### 3.5.1 Performance

### 3.5.2 Reliability

### 3.5.3 Availability

### 3.5.4 Security

### 3.5.5 Maintainability

### 3.5.6 Portability

## 3.6 Design Constraints

*Specify design constrains imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.*

## 3.7 Logical Database Requirements

*Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.*

## 3.8 Other Requirements

*Check all section for any additional requirements.*

# 4. Analysis Models <Should be completed in Assignment #3>

*List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.*

## 4.1 Sequence Diagrams

## 4.2 Data Flow Diagrams (DFD)

## 4.3 State-Transition Diagrams (STD)

# A. Appendices

*Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.*

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

## A.1 Appendix 1

## A.2 Appendix 2