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

02 – April – 2020

Hasan Haider

Inara Chagani

Zainab AliVyajkora

Raza Naqvi  
Muhammad Shahzain

Software Engineers

Prepared for

CS 353—Software Engineering

Instructor: UmairAzfar Khan.

Spring 2020

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 3/4/2020 | 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 7*

*3.3.2 Use Case #2 8*

*3.3.3 Use Case #3 9*

*3.3.4 Use Case #4 11*

*3.3.5 Use Case #5 12*

*3.3.5 Use Case #6 14*

*3.3.5 Use Case #7 15*

*3.3.8 Use Case #8 16*

*3.3.9 Use Case #9 17*

*3.3.10 Use Case #10 18*

*3.3.11 Use Case #11 19*

*3.3.12 Use Case #12 20*

*3.3.13 Use Case #13 21*

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.*

**Set user Preferences:** Based on the previous likes of the user, the system will show advertisements.

**Post Ads:** The User will be able to post advertisements on the selected websites.

**Fetch Ads:** The User will be able to search advertisements posted by people on the other websites.

**Create User Wish-list:** The User will be allowed to maintain a wish-list for the products they like.

## 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.*

2.3.1 A user may be an owner of a car that he wishes to sell. This could be a personal car owner or part of a car selling company or business.

2.3.2 A user may be a person looking to buy a car.

## 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.*

## 2.4.1 The accurateness of the ads being listed on the server depends on the time the user uses the website as the database is updated once at a fixed time every day.

2.4.2 The website will be available as a desktop version of a web application only. A mobile app does not exist.

2.4.3 The Scrap-Bot is entirely dependent upon the other websites from where it scrapes and crawls advertisements. Any change in the html tags of these websites will result in malfunction.

## 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.*

2.5.1 The user will be fluent in English language.

2.5.2 The user will have the hardware required to use the application, i.e. a machine, possibly keyboard/mouse/a tablet device.

2.5.3 The user will have a working browser to access the web application.

2.5.4 The system heavily depends on and assumes a stable network connection.

2.5.5 The system is dependent upon OLX, pakwheels, sastigari and pkmotors for fetching and posting advertisements.

# 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. It can be used on a web browser.

### 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 or any portable device with a screen and a browser available.

### 3.1.3 Software Interfaces

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

The web browser is essential to run the website. The websites mentioned in section 2.5.5 of this document must exist in the same way (in terms of html tags etc) for the system to work correctly.

### 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 signing 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.

* **Logout**

If at point prior to step 7 in the basic flow the Member selects to log out, 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**

* **Logout**

If at point prior to step 4 in the basic flow the Member selects to log out, 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

# The use case begins when User clicks on the “Update your Ad” button.

# Use Case: Update Advertisement is performed.

# The system prompts for car details.

# The system then asks the user to select the websites he/she wants to post the advertisement on.

# The user may then update previous credentials for the websites.

# The advertisement is posted.

# The use case ends successfully

# Alternative Flows

## Incomplete Car Details

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

## Logout

If at any point prior to step 7 in the basic flow the Member selects 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.

# 3.3.4 Use Case #4-Search Car Ads

# Brief Description

This use case describes how the member uses the website to search for a car advertisement using his/her account.

# Actors

## Member

## Scrapbot 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 Member clicks on "Search for Car Ads" button.
2. The System displays the different fields required to fill in the car specifications to search.
3. The User fills in the desired search criteria. [See reference document]
4. The Scrapbot returns matching results and the system displays all results based on the specified search criteria.
5. The User browses through the results.
6. The User selects a Car Ad.
7. The user adds the selected Car Ad to Wishlist.
8. The use case ends successfully.

# Alternative Flows

## No Search Criteria Specified

If in step 3 of the basic flow the User has not filled in any search criteria, then

1. The System shall display the message “Please specify Search criteria”.
2. The use case resumes at step 2.

## No Results Returned

If there are no results that match the search criteria specified by user in step 4, then

1. The System will display a message "No results found. Please enter new search criteria."
2. The use case resumes at step 2.

## No Car Selected

If in step 5 of the basic flow, Member does not select any of the car ads displayed, then

1. The User clicks on "New Search".
2. The use case resumes at step 2.

## Logout

If at point prior to step 8 in the basic flow the Member logs out, 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 user has selected a Car Ad and added it to Wishlist.

## Failure Condition

## The logs have been updated accordingly.

# Special Requirements

All steps are performed in a strict flow.

The database is actively updated in order to display most accurate information.

The user profile is maintained according to his/her search history and preferences.

# 3.3.5 Use Case #5-Send Message

# Brief Description

This use case describes how the member uses the website to send a message to another user using his/her account.

# Actors

## Member

## Scrapbot Database

# Preconditions

There is an active network connection to the website.

Both the User and the message recipient already have an account on the website.

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

The Recipient of the message is added as a contact in the User's messenger account or reached directly through an Ad.

The User must have completed one of the Use cases "Show Result" or "Browse Wishlist".

# Basic Flow of Events

# The use case begins when Member clicks on "Contact Seller" button on an Ad or selects an already added User in his/her messenger contacts.

# The System prompts the User to type in a message.

# The User types in his/her message.

# The User clicks on the "Send" button.

# The System prompts the user to confirm whether he/she wants to send the message.

# The User selects "Confirm".

# The System displays a message "Message successfully sent".

# The use case ends successfully.

# Alternative Flows

## Message field empty

If in step 3 of the basic flow the User has left the Message field empty, then

1. the System shall display the message “Please type a message. The message cannot be empty.”.
2. The use case resumes at step 2.

## No Confirmation

If in step 6 the user selects "Go Back", then

1. The System will display the message the user has typed and he may edit it.
2. The use case resumes at step 3.

## Logout

If at any point prior to step 8 in the basic flow the Member selects to log out, 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 user has sent a message to the seller/other user.

## Failure Condition

The logs have been updated accordingly.

# Special Requirements

All steps are performed in a strict flow.

The user profile and messenger history is updated and new contacts are added to the list of active conversations.

# 3.3.6 Use Case #6-Browse Wishlist

# Brief Description

This use case describes how member browses wishlist and performs different actions on advertisements in wishlist .

# Actors

# Member

# Scrapbot database

# Preconditions

There is an active network connection to the website.

# Basic Flow of Events

1. The use case begins when the member clicks “Browse Wishlist” button.
2. Advertisements in the wishlist are displayed.
3. The User may delete any advertisement from the wishlist.
4. The User may send message to any seller whose advertisements are displayed.
5. The use case ends successfully.

# Alternative Flows

## No advertisements in the wishlist

In step 2 of basic flow of events, if wishlist is empty then

1. No Advertisements are displayed and step 3 and 4 of basic flow of events are no longer valid.

# Key Scenarios

## No Response from Website

# Post-conditions

## Successful Completion

The user has successfully viewed advertisements from the wishlist. If user has removed any advertisement from wishlist, it has been removed. If user has contact any seller, his message has been sent to the seller.

## Failure Condition

The logs have been updated accordingly.

# Special Requirements

The database is updated only if any advertisement is removed from the wishlist or a seller has been sent message to by the user.

# 3.3.7 Use Case #7-Scrap Car from websites

# Brief Description

This use case describes how the system scraps car advertisements from different advertisement websites like PakWheels, olx etc.

# Actors

## Scrapbot Database

# Preconditions

The system has the URLs of all the car advertisement websites.

The system has sufficient storage capacity.

System has an active network connection.

# Basic Flow of Events

1. The use case begins automatically at a fixed time everyday.
2. The system extracts the URLs of all car advertisement websites.
3. The system accesses all the advertisement websites and scraps all car advertisements posted on these websites. Scrapped advertisements are stored on the database.
4. All duplicate advertisements are removed from the database.
5. Finally, the database is updated and use case ends.

# Alternative Flows

## Website Error

In step 3 of basic flow of events, if any advertisement website cannot be accessed then

* 1. Advertisements will not be scrapped from websites with access issues.
  2. The use case will end with failure condition.

# Key Scenarios

## System failure

# Post-conditions

## Successful Completion

Recently posted car advertisements are available on scrapbot’s website.

## Failure Condition

The logs have been updated accordingly.

# Special Requirements

In case of layout modification of an advertisement website, new code must be written to scrap advertisements from that website.

# 3.3.8 Use Case #8 Login

# Brief Description

This use case describes how the Bank Customer uses the website to login to his/her account.

# Actors

## Member

## Scrap-bot Database

# Preconditions

There is an active network connection to the website.

The User already has an account

# Basic Flow of Events

1. The use case begins when the User clicks the “Log In” button.
2. Use Case: “Login” is performed.
3. The system prompts the user to fill in his/her username and password.
4. The user will then press the “login” button to verify his/her credentials.
5. The user has successfully logged in.
6. The use case ends successfully

# Alternative Flows

## Invalid Login

If in step 4 of the basic flow Member the use case: Verify Credentials does not complete successfully, then

1. the web page shall display the message “Invalid Username/Password – please try again”
2. The use case resumes at step 3

## Logout

If at point prior to step 6 in the basic flow the User logs out, then

1. The use case ends

# Key Scenarios

## No Response from Website

# Post-conditions

## Successful Completion

The Member has successfully logged in.

## 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.

# 3.3.9 Use Case #9 Sign Up

# Brief Description

This use case describes how the Member uses the website to create an account.

# Actors

## Member

## Scrap-Bot Database

# Preconditions

There is an active network connection to the website.

# Basic Flow of Events

1. The use case begins when the User clicks on the “Sign Up” button.
2. Use Case: “Signup” is performed.
3. The system prompts the User to fill in the details.
4. The user will then press the “Sign Up” button to validate user information.
5. The user successfully signs up.
6. The use case ends successfully.

# Alternative Flows

## User already exists.

If in step 4 of the basic flow, the Member account already exists, then

1. The web page shall display the message, “This Member already exists”.
2. The use case ends.

## Invalid Value

If in step 4 of the basic flow, the Member enters an invalid value in any of the data fields, then

1. The web page will display the message “Invalid Value in this field!”, and ask the user to enter a valid value.
2. The use case resumes at step 3

## Logout

If at a point prior to step 6 in the basic flow the Member logs out, then

1. The system prompts the user that the account has not been created.
2. The use case ends.

# Key Scenarios

## No Response from Website

# Post-conditions

## Successful Completion

The member account has been created 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 member account has been created.

# 3.3.10 Use Case #10 View Messages

# Brief Description

This use case describes how the member uses the website to view all his/her messages.

# Actors

## Member

## Scrap-Bot Database

# Preconditions

There is an active network connection to the website.

The user is already logged in.

# Basic Flow of Events

1. The use case begins when User clicks on the “Chat” tab.
2. Use Case: “View Messages” is performed.
3. The system displays all the messages, from various sellers, to the user.
4. The user has the liberty to search specific messages.
5. The use case ends successfully

# Alternative Flows

## Logout

If at a point prior to step 4 in the basic flow the User logs out, then

1. The use case ends.

# Key Scenarios

## No Response from Website

# Post-conditions

## Successful Completion

The system displays all the messages.

## 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.

# 3.3.11 Use Case #11 View Notifications

# Brief Description

This use case describes how the member uses the website to view all his/her notifications.

# Actors

## Member

## Scrap-Bot Database

# Preconditions

There is an active network connection to the website.

The user is already logged in.

# Basic Flow of Events

1. The use case begins when User clicks on the “View Notifications” tab.
2. Use Case: “View Notifications” is performed.
3. The system displays all the notifications to the user.
4. The use case ends successfully

# Alternative Flows

## Logout

If at a point prior to step 4 in the basic flow the User logs out, then

1. The use case ends.

# Key Scenarios

## No Response from Website

# Post-conditions

## Successful Completion

The system displays all the notifications.

## 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.

# 3.3.12 Use Case #12Set Preferences

# Brief Description

This use case describes how the member uses the website to set his/her car preferences.

# Actors

## Member

## Scrap-Bot Database

# Preconditions

There is an active network connection to the website.

The user is already logged in.

# Basic Flow of Events

1. The use case begins when User clicks on the “Set Preferences” button.
2. Use Case: “Set Preferences” is performed.
3. The system prompts the User to fill in his/her desired car preferences and features.
4. The user will then press the “Save” button to store the preferences.
5. The data is saved in the database.
6. The use case ends successfully

# Alternative Flows

## Log out

If at a point prior to step 6 in the basic flow the User logs out, then

1. The system prompts the user that any preferences selected were not saved.
2. The use case ends.

# Key Scenarios

## No Response from Website

# Post-conditions

## Successful Completion

The database has been successfully updated with the preferences selected by the user.

## 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 user saves the preferences.

# 3.3.13 Use Case #13 Edit User Profile

# Brief Description

This use case describes how the member uses the website to edit his/her profile.

# Actors

## Member

## Scrap-Bot Database

# Preconditions

There is an active network connection to the website.

The user is already logged in.

# Basic Flow of Events

1. The use case begins when User clicks on the “Edit Profile” button.
2. Use Case: “Edit User Profile” is performed.
3. The system prompts the User to update some details.
4. The user will then press “Save changes” button to validate entered information.
5. The updates are saved in the database.
6. The use case ends successfully

# Alternative Flows

## Invalid Value

If in step 4 of the basic flow, the Member enters an invalid value in any of the data fields, then

1. The web page will display the message “Invalid Value in this field!”, and ask the user to enter a valid value.

2. The use case resumes at step 3

## Logout

If at a point prior to step 6 in the basic flow the User chooses to log out, then

1. The system prompts the user that any updates made were not saved.
2. The use case ends

# Key Scenarios

## No Response from Website

# Post-conditions

## Successful Completion

The database has been successfully updated with the relevant changes made by the user.

## 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 user saves the changes.

## 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).*

### Performance:

Depends on the quality of Internet. Ideally, any transaction should not take more than a second. 90% of the transactions shall be done in less than a second. The system shall not crash unless the browser crashes. The server shall be able to handle around 99 requests per second.

### Reliability:

If the website crashes, the user will know in less than a second. If a transaction cannot be completed, the system will inform in less than a second. The probability of system failure is 1/100 approximately.

### Availability:

Since it is a website, it will be available 24/7. The website (or parts of website) may be unavailable during maintenance.

### Security:

The User has an account and their data cannot be accessed without the credentials.

### Maintainability:

The website can be maintained while people are still using it. It will only go under maintenance if bugs are found.

### Portability:

With the availability of internet and a browser, the website can be access on laptops, desktop systems, mobile phones, tablets and any other devices anywhere in the world.

## 3.6 Design Constraints

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

Not Applicable.

## 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.*

SQL Server database will be used.

## 3.8 Other Requirements

*Check all section for any additional requirements.*

Not Applicable.

# 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